NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

NGA.SIG.0004_1.1
2011-08-01

NGA STANDARDIZATION DOCUMENT

Light Detection and Ranging (LIDAR) Sensor Model

Supporting Precise Geopositioning

(2011-08-01)

Version 1.1

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Table of Contents

Table of Figures

Revision History

Version Identifier	Date	Revisions/notes
0.0.1	07 July 2009	Final edit for review / comment
1.0	30 November 2009	Final rework based on comments received during review/comment period.
1.1	August 2011	Version 1.1 revision adds “On-demand” and “Pre-computed” error model implementations, as well as justification for an error model, edits to the Key Components and CSM functions, application of the model for determination of estimated error, descriptions of model exploitation using the on-demand and pre-computed implementations, and additions to the metadata tables. . This version also includes edits to descriptive figures, symbols for clarity and consistency, modifications to equations for consistent units and symbols, additional description of CSM functions, and various grammatical / format errors throughout.

1. Introduction

1.1 Background/Scope

The National Geospatial-Intelligence Agency (NGA), National Center for Geospatial Intelligence Standards (NCGIS) and the Geospatial Intelligence Standards Working Group (GWG) engaged with the Department of Defense components, the Intelligence Community, industry and academia in an endeavor to standardize descriptions of the essential sensor parameters of collection sensor systems by creating "sensor models."

The goal and intent of this information and guidance document is to detail the sensor and collector physics and dynamics enabling equations to establish the geometric relationship among sensor, image, and object imaged. This document was developed to complement existing papers for frame imagery, spotlight SAR and whiskbroom/Pushbroom, which are published and under configuration management. This document migrates from the traditional 2-D image to a 3-D range "image" scenario. Its primary focus is upon airborne topographic LIDAR and includes both frame and point scanning systems. However, the basic principals could be applied to other systems such as airborne bathymetric systems or ground-based / terrestrial systems. The document promotes the validation and Configuration Management (CM) of LIDAR geopositioning capabilities across the National System for Geospatial-Intelligence (NSG) to include government- / military-developed systems and Commercial-of-the-Shelf (COTS) systems.

The most significant change made in this version was the addition of the "On-demand" and "Pre-computed" error model implementations. This includes a justification for an error model (Section 5), edits to the Key Components and CSM functions (Section 6.1), application of the model for determination of estimated error (Section 6.3), descriptions of model exploitation using the on-demand and pre-computed implementations (Appendices A and B), and additions to the metadata tables (Section 7). This version also includes many suggested changes provided by community as a result of the review/comment period. Included are edits to and the addition of descriptive figures (Sections 2, 3 and 4), edits to the symbols for clarity and consistency, modifications of equations for consistent units and symbols (Section 4), additional description of the CSM functions (Section 6), and various typographical, grammatical, and format errors throughout the document.

The decision to publish this version was made in full consideration and recognition that additional information is being developed on a daily basis. The community requirement for information sharing and continued collaboration on LIDAR technologies justifies going ahead with this release.

The reader is advised that the content of this document represents the completion of a second development and review effort by the development team. With the publication of this version, actions have been initiated to place it under strict configuration management and continue a peer review process for change and update. The reader is cautioned that inasmuch as the development process is on-going, all desired/necessary changes may not have been incorporated into the second release. When possible, the development team has noted areas that are known to be in flux. The reader is encouraged to seek additional technical advice and/or assistance from the Community Sensor Model Working Group (CSMWG), or the NGA Sensor Geopositioning Center (SGC).

Illustrative of the work that needs to be addressed is determining the relationship between the NGA InnoVision Conceptual Model and Metadata Dictionary (CMMD) for LIDAR and the LIDAR Formulation Paper. The CMMD is currently addressing many aspects of LIDAR metadata to include the geopositioning. Should metadata tables in the Formulation Paper be removed and instead reference the very extensive CMMD? Some of the comments received from the community remain unanswered pending a decision on this question.

Finally, collaboration will continue with the community to ensure that the document reflects current LIDAR collection and processing techniques

1.2 Approach

This technical document details various parameters to consider when constructing a sensor model. It focuses on two primary classes of LIDAR sensors: frame scanning sensors and point scanning sensors. A frame-scanner is a sensor that acquires all of the data for an image (frame) at an instant of time. Typical of this class of sensor is that it has a fixed exposure and is comprised of a two-dimensional detector or array, such as a Focal Plane Array (FPA) or Charge-Coupled Device (CCD) array. A point-scanner is a sensor that acquires data for one point (or pixel) at an instant of time. A point-scanner can be considered a frame-scanner of 1 pixel in size.

LIDAR systems are very complex and although there are some “standardized” COTS systems, individual systems generally have very unique properties. It would be impossible for this paper to capture the unique properties of each system. Therefore, the focus of this paper will be on those generalized geometric sensor properties necessary for accurate geolocation with frame-scanning and point-scanning sensors. These generalized parameters will need to be modified for implementation on specific systems, but the basic framework developed in this paper will still apply. The goal of this paper is to lay out the principles that can then be applied as necessary. Additionally, relationships other than geometric (e.g. spectral) are known to exist, but are beyond the scope of this paper.

1.3 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Community Sensor Model (CSM) Technical Requirements Document, Version 3.0, December 15, 2005.

Federal Geographic Data Committee (FGDC) Document Number FGDC-STD-012-2002, Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata.

North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG), Air Reconnaissance Primary Imagery Data Standard, Base document STANAG 7023 Edition 3, June 29, 2005.

1.4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

1.4.1. **adjustable model parameters**

model parameters that can be refined using available additional information such as ground control points, to improve or enhance modeling corrections

1.4.2. **attitude**

orientation of a body, described by the angles between the axes of that body's coordinate system and the axes of an external coordinate system [ISO 19116]

1.4.3. **area recording**

"instantaneously" recording an image in a single frame

1.4.4. **attribute**

named property of an entity [ISO/IEC 2382-17]

1.4.5. **calibrated focal length**

distance between the projection center and the image plane that is the result of balancing positive and negative radial lens distortions during sensor calibration

1.4.6. **coordinate**

one of a sequence of *n* numbers designating the position of a point in *n*-dimensional space [ISO 19111]

NOTE: In a coordinate reference system, the numbers must be qualified by units.

1.4.7. **coordinate reference system**

coordinate system that is related to the real world by a datum [ISO 19111]

NOTE: A geodetic or vertical datum will be related to the Earth.

1.4.8. **coordinate system**

set of mathematical rules for specifying how coordinates are to be assigned to points [ISO 19111]

1.4.9. **data**

reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or processing [ISO/IEC 2382-1]

1.4.10. **error propagation**

determination of the covariances of calculated quantities from the input covariances of known values

1.4.11. **field of view**

The instantaneous region seen by a sensor provided in angular measure. In the airborne case, this would be swath width for a linear array, ground footprint for an area array, and for a whisk broom scanner it refers to the swath width. [Manual of Photogrammetry]

1.4.12. field of regard

The possible region of coverage defined by the FOV of the system and all potential view directions of the FOV enabled by the pointing capabilities of the system, i.e. the total angular extent over which the FOV may be positioned. [adapted from the Manual of Photogrammetry]

1.4.13. first return

For a given emitted pulse, it is the first reflected signal that is detected by a 3-D imaging system, time-of-flight (TOF) type, for a given sampling position [ASTM E2544-07a]

1.4.14. frame

The data collected by the receiver as a result of all returns from a single emitted pulse.

A complete 3-D data sample of the world produced by a LADAR taken at a certain time, place, and orientation. A single LADAR frame is also referred to as a *range image*. [NISTIR 7117]

1.4.15. frame sensor

sensor that detects and collects all of the data for an image (frame / rectangle) at an instant of time

1.4.16. geiger mode

LIDAR systems operated in a mode (photon counting) where the detector is biased and becomes sensitive to individual photons. These detectors exist in the form of arrays and are bonded with electronic circuitry. The electronic circuitry produces a measurement corresponding to the time at which the current was generated; resulting in a direct time-of-flight measurement. A LADAR that employs this detector technology typically illuminates a large scene area with a single pulse. The direct time-of-flight measurements are then combined with platform location / attitude data along with pointing information to produce a three-dimensional product of the illuminated scene of interest. Additional processing is applied which removes existing noise present in the data to produce a visually exploitable data set. [adapted from Albota 2002]

1.4.17. geodetic coordinate system

coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height [ISO 19111]

1.4.18. geodetic datum

datum describing the relationship of a coordinate system to the Earth [ISO 19111]

NOTE 1: In most cases, the geodetic datum includes an ellipsoid description
NOTE 2: The term and this Technical Specification may be applicable to some other celestial bodies.

1.4.19. geographic information

information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19101]

1.4.20. geographic location

longitude, latitude and elevation of a ground or elevated point

1.4.21. geolocating

geopositioning an object using a sensor model

1.4.22. geopositioning

determining the ground coordinates of an object from image coordinates

1.4.23. ground control point

point on the ground, or an object located on Earth surface, that has accurately known geographic location

1.4.24. image

coverage whose attribute values are a numerical representation of a remotely sensed physical parameter

NOTE: The physical parameters are the result of measurement by a sensor or a prediction from a model.

1.4.25. image coordinates

coordinates with respect to a Cartesian coordinate system of an image

NOTE: The image coordinates can be in pixel or in a measure of length (linear measure).

1.4.26. image distortion

deviation in the location of an actual image point from its theoretically correct position according to the geometry of the imaging process

1.4.27. image plane

plane behind an imaging lens where images of objects within the depth of field of the lens are in focus

1.4.28. image point

point on the image that uniquely represents an object point

1.4.29. imagery

representation of objects and phenomena as sensed or detected (by camera, infrared and multispectral scanners, radar and photometers) and of objects as images through electronic and optical techniques [ISO/TS 19101-2]

1.4.30. instantaneous field of view

The instantaneous region seen by a single detector element, measured in angular space. [Manual of Photogrammetry]

1.4.31. intensity

The power per unit solid angle from a point source into a particular direction. Typically for LIDAR, sufficient calibration has not been done to calculate absolute intensity, so relative intensity is usually reported. In linear mode systems, this value is typically provided as an integer, resulting from a mapping of the return's signal power to an integer value via a lookup table.

1.4.32. LADAR

Acronym for **La**ser **D**etection **a**nd **R**anging, or Laser Radar. This term is used interchangeably with the term LIDAR. (Historically, the term LADAR grew out of the Radar community and is more often found in the literature to refer to tracking and topographic systems.)

1.4.33. last return

For a given emitted pulse, it is the last reflected signal that is detected by a 3-D imaging system, time-of-flight (TOF) type, for a given sampling position [reference ASTM E2544-07a]

1.4.34. LIDAR

Acronym for **Li**ght **D**etection **a**nd **R**anging. A system consisting of 1) a photon source (frequently, but not necessarily a laser), 2) a photon detection system, 3) a timing circuit, and 4) optics for both the source and the receiver that uses emitted light to measure ranges to and/or properties of solid objects, gases, or particulates in the atmosphere. Time-of-flight (TOF) LIDARs use short laser pulses and precisely record the time each laser pulse was emitted and the time each reflected return(s) is received in order to calculate the distance(s) to the scatterer(s) encountered by the emitted pulse. For topographic LIDAR, these time-of-flight measurements are then combined with precise platform location/attitude data along with pointing data to produce a three-dimensional product of the illuminated scene of interest.

1.4.35. linear mode

LIDAR systems operated in a mode where the output photocurrent is proportional to the input optical incident intensity. A LIDAR system which employs this technology typically uses processing techniques to develop the time-of-flight measurements from the full waveform that is reflected from the targets in the illuminated scene of interest. These time-of-flight measurements are then combined with precise platform location / attitude data along with pointing data to produce a three-dimensional product of the illuminated scene of interest. [adapted from Aull, 2002]

1.4.36. metadata

data about data [ISO 19115]

1.4.37. multiple returns

For a given emitted pulse, a laser beam hitting multiple objects separated in range is split and multiple signals are returned and detected [reference ASTM E2544-07a]

1.4.38. nadir

The point of the celestial sphere that is directly opposite the zenith and vertically downward from the observer (Merriam-Webster Online Dictionary)

1.4.39. object point

point in the object space that is imaged by a sensor

NOTE: In remote sensing and aerial photogrammetry an object point is a point defined in the ground coordinate reference system.

1.4.40. objective

optical element that receives light from the object and forms the first or primary image of an optical system

1.4.41. pixel

picture element [ISO/TS 19101-2]

1.4.42. point cloud

A collection of data points in 3D space. The distance between points is generally non-uniform and hence all three coordinates (Cartesian or spherical) for each point must be specifically encoded.

1.4.43. platform coordinate reference system

coordinate reference system fixed to the collection platform within which positions on the collection platform are defined

1.4.44. principal point of autocollimation

point of intersection between the image plane and the normal from the projection center

1.4.45. projection center

point located in three dimensions through which all rays between object points and image points appear to pass geometrically

NOTE: It is represented by the near nodal point of the imaging lens system.

1.4.46. pulse repetition frequency

number of times the LIDAR system emits pulses over a given time period, usually stated in kilohertz (kHz)

1.4.47. receiver

Hardware used to detect and record reflected pulse returns. A general laser radar receiver consists of imaging optics, a photosensitive detector (which can have one to many elements), timing circuitry, a signal processor, and a data processor. The receiver may be such that it detects only one point per epoch, or an array of points per epoch.

1.4.48. remote sensing

collection and interpretation of information about an object without being in physical contact with the object

1.4.49. return

A sensed signal from an emitted laser pulse which has reflected off of an illuminated scene of interest. There may be multiple returns for a given emitted laser pulse.

1.4.50. scan

One instance of a scanner’s repeated periodic pattern.

1.4.51. sensor

element of a measuring instrument or measuring chain that is directly affected by the measurand [ISO/TS 19101-2]

1.4.52. sensor model

mathematical description of the relationship between the three-dimensional object space and the associated sensing system.

1.4.53. swath

The ground area from which return data are collected during a continuous airborne LIDAR operation. A typical mapping mission may consist of multiple adjacent swaths, with some overlap, and the operator will turn off the laser while the aircraft is oriented for the next swath. This term may also be referred to as a pass.

1.4.54. swipe

The set of sequential frames collected during a single half-cycle of a mechanical scanner representing a cross-track excursion from one side of the field of regard to the other

1.4.55. topographic LIDAR

LIDAR systems used to measure the topography of the ground surface and generally referring to an airborne LIDAR system

1.4.56. voxel

A volume element, the 3D equivalent to a pixel in 2D.

1.5 Symbols and abbreviated terms

1.5.1 Abbreviated terms

ANSI	American National Standards Institute
APD	Avalanche Photo Diode (ATM E2544-07a)
CCD	Charge Coupled Device
CONOP	Concept of Operations
ECEF	Earth Centered Earth Fixed
ENU	East North Up
FOV	Field of View
FOR	Field of Regard
FPA	Focal Plane Array
GmAPD	Geiger-mode Avalanche PhotoDiode
GPS	Global Positioning System
IFOV	Instantaneous Field of View
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LADAR	Laser Detection and Ranging System
LIDAR	Light Detection and Ranging System
NED	North East Down
PRF	Pulse Repetition Frequency
PPS	Pulses per second
TOF	Time-of-Flight

1.5.2 Symbols

A	object point coordinate (ground space)
a	Image vector
a_1, b_1, c_1, a_2, b_2, c_2	parameters for a six parameter transformation, in this case to account for array distortions
c	speed of light
c	column in the row-column coordinate system
C_l	Line correction for row-column to line-sample conversion
C_s	Sample correction for row-column to line-sample conversion
D	Down in the North East Down (NED) Coordinate System
E	East in the North East Down (NED) or East North Up (ENU) Coordinate System
f	camera focal length
H	Heading in reference to the local-vertical coordinate system
H	flying height above mean sea level (MSL) of the aircraft, in kilometers
h	height above MSL of the object the laser intersects, in kilometers

i	index of frames
j	index of points
K	refraction constant, micro-radians
k	arbitrary constant
k_1, k_2, k_3	first, second, and third order radial distortion coefficients, respectively
L	front nodal point of lens
ℓ	line in the line sample coordinate system
M_{ECEF}	rotation matrix from the ellipsoid-tangential (NED) reference frame to the ECEF reference frame
M_{ELL}	rotation matrix from the local-vertical reference frame to the ellipsoid-tangential reference frame
M_{GIM}	rotation matrix from the sensor reference frame to the gimbal reference frame (gimbal angles)
M_{PLA}	rotation matrix from the gimbal reference frame to the platform reference frame (boresight angles)
M_{SEN}	rotation matrix from scanner reference frame to sensor reference frame (scan angles)
M_{VER}	rotation matrix from the platform reference frame to the local-vertical reference frame (IMU observations)
$\mathbf{M}$	rotation matrix (various)
$\mathbf{M}_{\omega}$	rotation matrix about the x-axis (roll)
$\mathbf{M}_{\varphi}$	rotation matrix about the y-axis (pitch)
$\mathbf{M}_{\kappa}$	rotation matrix about the z-axis (yaw)
$\mathbf{M}$	the orientation matrix
N	North in the North East Down (NED) or North East Up Coordinate System
P	Pitch in reference to the local-vertical coordinate system
p_1, p_2	lens decentering coefficients
r	radial distance on image from principal point to point of interest
r_{ECEF}	vector from the ECEF origin to the GPS antenna phase-center in the ECEF reference frame (GPS observations)
r_{EP}	vector from the ECEF origin to the ground point in the ECEF reference frame
r_{GIM}	vector from the sensor to the gimbal center of rotation in the gimbal reference frame
r_{GPS}	vector from the GPS antenna phase-center to the IMU in the platform reference frame
r_{INSIMU}	vector from the IMU to the gimbal center of rotation in the platform reference frame
r_{SCA}	vector from the scanner to the ground point in the scanner reference frame (range)
R	Roll in reference to the local-vertical coordinate system
R	range
R’	range from front nodal point (L) to the point on the ground (A)
r	row in the row-column coordinate system
s	sample in the line-sample coordinate system
T	period of Signal
t	round trip travel time
x	x coordinate in the x-y frame coordinate system
$\bar{x}, \bar{y}$	image coordinates adjusted by principal point offset
X,Y,Z	right-handed Cartesian ground coordinate system
X_a, Y_a, Z_a	Cartesian Coordinates in Local-Vertical Coordinate System
$X_LY_LZ_L$	the position of the sensor front nodal point in ground coordinates
X_W,Y_W, Z_W	Cartesian Coordinates in World Coordinate System

x'	image coordinate (x-component) adjusted for lens and atmospheric errors
x_{GIM}, y_{GIM}, z_{GIM}	Cartesian Coordinates in Gimbal Coordinate System
X_0, Y_0	Principal point offset in the x-y frame coordinate system
x_p, y_p, z_p	Cartesian Coordinates in Platform Coordinate System
x_s, y_s, z_s	Cartesian Coordinates in Sensor Coordinate System
x_{sc}, y_{sc}, z_{sc}	Cartesian Coordinates in Scanner Coordinate System
y	y coordinate in the x-y frame coordinate system
y'	image coordinate (y-component) adjusted for lens and atmospheric errors
ϕ	Phase shift of signal
Φ	Latitude
λ	Longitude
α	the angle of the laser beam from vertical
$\dot{B}_{ij}$	coefficients of the unknown corrections to the sensor parameters
$\bar{B}_{ij}$	coefficients of the unknown corrections to the ground coordinates
Δd	angular displacement of the laser beam from the expected path
Δx_{atm}	atmospheric correction for image coordinates, x-component
Δy_{atm}	atmospheric correction for image coordinates, y-component
Δx_{dec}	lens decentering errors, x component
Δy_{dec}	lens decentering errors, y component
Δx_{lens}	total lens radial distortion and decentering distortion, x-component
Δy_{lens}	total lens radial distortion and decentering distortion , y-component
Δx_{radial}	radial lens distortions, x-component
Δy_{radial}	radial lens distortions, y-component
$\dot{\mathbf{\Delta}}_i$	unknown corrections to the sensor parameters
$\bar{\mathbf{\Delta}}_j$	unknown corrections to the ground coordinates
θ	angular displacement of the range vector from the lens optical axis, x-component
φ	angular displacement of the range vector from the lens optical axis, y-component
s	adjustment for the range to account for distance from front nodal point to the lens
v_{ij}	residuals of the frame coordinates
φ	pitch
ω	roll
κ	yaw

2. LIDAR Overview

2.1 Overview of LIDAR Sensor Types

2.1.1. Introduction

Light **D**etection **A**nd **R**anging (LIDAR) refers to a radar system operating at optical frequencies that uses a laser as its photon source (Kamerman). There are many varieties of LIDAR in operation, performing different missions. Some systems like the Scanning Hydrographic Operational Airborne LIDAR Survey (SHOALS) and the Compact Hydrographic Airborne Rapid Total Survey (CHARTS) deploy LIDAR using wavelengths that are optimal for collecting shallow bathymetry and other data needed for detecting obstacles to navigation. Others, including the majority of the COTS systems, such as the Optech 3100 and the Leica ALS50, are focused on topographic mapping and used to make a map or 3D image of locations on the earth. There are still other LIDAR systems used in completely different applications, such as the detection of gases. This paper focuses on topographic LIDAR systems, or systems used to make a map or 3D image of the area of interest. This document, or similar documents, may be expanded in the future to address other applications of LIDAR systems.

Topographic LIDAR systems generally measure the travel time, time between a laser pulse emission and when the reflected return is received, and use this to calculate the range (distance) to the objects encountered by the emitted pulse. By combining a series of these ranges with other information such as platform location, platform attitude and pointing data, a three dimensional (3D) scene of the area of interest is generated. Often this scene is stored as a series of 3D coordinates, {X,Y,Z}, per return that is called a point cloud. Many variations of LIDAR systems have been developed. This paper provides a general overview of the technology and gives the reader enough insight into the technology to understand the physical sensor model described later in this document. For additional information on LIDAR technologies, the reader is encouraged to read the papers and texts referenced in this document.

2.1.2. System Components

Although there are many variants of LIDAR systems, these systems generally consist of a similar set of core components that include: ranging subsystem (laser transmitter, laser receiver), scanning/pointing subsystem, position and orientation subsystem, system controller, and data storage (Brenner, Liadsky, and Wehr). All of these components are critical to the development of a 3D dataset. Additionally, when developing the physical model of the LIDAR system, many of these components have their own coordinate systems as detailed later in this document. Each of these core components of LIDAR systems are shown in Figure 1 and described below.

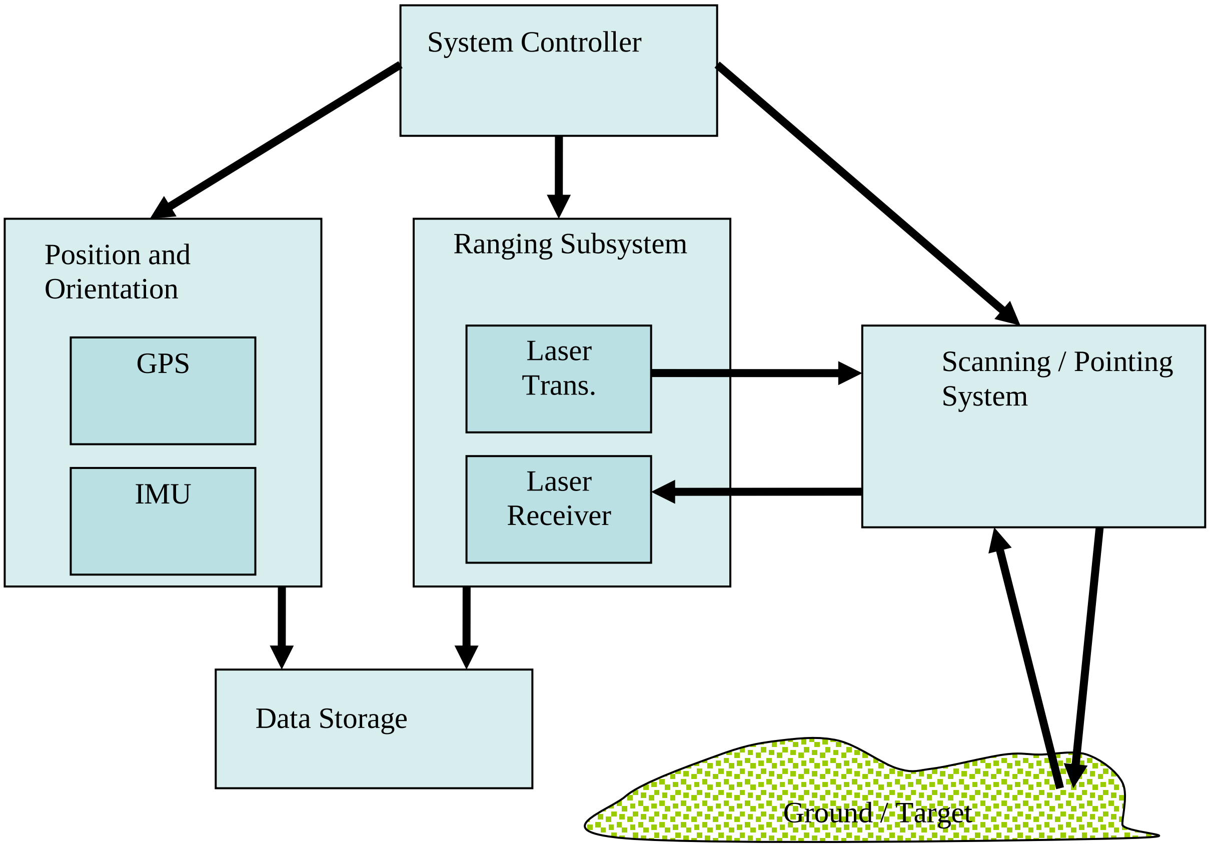


Figure 1. LIDAR Components

2.1.2.1. Ranging Subsystem

The key component that defines LIDAR as a unique system is the ranging subsystem. This system consists of additional subsystems including a laser transmitter and an electro-optical receiver.

The laser transmitter generates the laser beam and emits the laser energy from the system which is then pointed toward the ground by other subsystems. There can be multiple components along the optical path of the laser energy as it is transmitted, including a transmit-to-receive switch, beam expanders, and output telescope optics to name a few (Kamerman). There are multiple laser types that could be used for LIDAR systems with one common type being neodymium-doped yttrium aluminum garnet (Nd:YAG). LIDAR systems are operated at a variety of wavelengths with the most common being 1064 nm (near infrared) for topographic scanners and 532 nm (green) for bathymetric scanners. Terrestrial scanners often use higher wavelengths (~1500 nm) for maximizing eye safety. The selection of the laser wavelength depends upon a variety of factors including: the characteristic of the environment being measured, the overall system design, the sensitivity of the detectors being used, eye safety, and the backscattering properties of the target (Wehr). In addition to the laser wavelength, the laser power is also an important consideration in relation to eye safety.

The electro-optical receiver captures the laser energy that is scattered or reflected from the target and focuses the energy onto a photosensitive detector using the imaging optics. Timestamps from the transmitted and detected light are then used to calculate travel time and therefore range.

2.1.2.1.1. Ranging Techniques

For LIDAR, one of two ranging principles is usually applied: pulsed ranging or continuous wave.

In pulsed modulated ranging systems, also known as time-of-flight, the laser emits single pulses of light in rapid succession (Pulse Repetition Frequency – PRF). The travel time between the pulse being emitted and then returning to the receiver is measured. This time, along with the speed of light can be used to calculate the range from the platform to the ground:

$$Range = R = \frac{1}{2} c \cdot t$$

Eq. 1

Where: c = speed of light and t = round trip travel time

In continuous wave systems, the laser transmits a continuous signal. The laser energy can then be sinusoidally modulated in amplitude and the travel time is directly proportional to the phase difference between the received and the transmitted signal. This travel time is again used with the speed of light to calculate the range from the sensor to the ground.

$$t = \frac{\phi}{2\pi} \cdot T \text{ where } \phi \text{ is the phase shift and } T \text{ is the period of the signal}$$

Eq. 2

Once the travel time (t) is known, the range is calculated as indicated above. To overcome range ambiguities, multiple-tone sinusoidal modulation can be used, where the lowest frequency tone has an ambiguity greater than the maximum range of the system (Kamerman).

An alternative method in continuous wave systems would involve modulation in frequency. These *chirped* systems would mix the received signal with the transmitted signal and then use a coherent receiver to demodulate the information encoded in the carrier frequency (Kamerman).

Note that Equations Eq. 1 and Eq. 2, as shown above, presume that the sensor is stationary during sensing. Some sensor applications may need to account for sensor movement during sensing. This paper does not provide examples of accounting for this movement.

2.1.2.1.2. Detection Techniques

There are two detection techniques generally employed in LIDAR detection systems. These are direct detection and coherent detection. In one form of direct detection, referred to as linear mode, the receiver converts the return directly to a voltage or current that is proportional to the incoming optical power. Possible receivers include Avalanche Photo Diodes (APD) and Photo Multiplier Tubes (PMT).

LIDAR detectors (APDs and others) can also be operated in a photon counting mode. When photon counting, the detector is sensitive to very few and possibly individual photons. In a Geiger mode photon counting system, the detector is biased to become sensitive to individual photons. The electronic circuitry associated with the receiver produces a measurement corresponding to the time that a current is generated from an incoming photon, resulting in a direct time-of-flight measurement. (Albota 2002)

In coherent detection the received optical signal is mixed with a local oscillator through a heterodyne mixer prior to being focused on the photosensitive element. The mixing operation converts the information to a narrow base band which reduces the noise signal as compared with the optical filter employed in the direct detection approach. The resultant SNR improvement can be substantial as in the case with atmospheric turbulence detection systems.

In addition to the methods described above, some systems are using alternative detection techniques. One such technique uses the polarization properties of the energy to determine range. As this paper is meant to focus on the LIDAR geometric sensor model, it will not discuss all possible ranging and detection techniques.

2.1.2.1.3. Flying Spot versus Array

The sections above described both ranging techniques and detection techniques that are used in laser scanning. However, it is important to note that these techniques lead to various receiver geometries for collecting the data. In general, most commercial LIDAR systems operate on a flying spot principle where for a single outgoing pulse, a small number of ranges (between 1 and 5) are recorded for the returning energy along the same line of sight vector. Receiving and recording more than one range for a given pulse is often referred to as Multiple Returns. The first range measured from a given pulse is often referred to as the “First Return” and the last as the “Last Return”. For the next pulse, the pointing system has changed the line of sight vector, and an additional small number of ranges are recorded. This method (point scanning) is generally associated with linear-mode systems where the energy is focused on a small area on the ground and a large return signal is required to record a return and calculate a range. However, there are other systems (photon counting and others) that spread higher power outgoing energy to illuminate a larger area on the ground and use a frame array detector to measure a range for each pixel of the array. These systems (frame scanning) require low return signal strength and record hundreds or even thousands of ranges per outgoing pulse. There are pros and cons to both systems which will not be debated in this document. However, it is important that the reader realize that both point scanning and frame scanning LIDAR systems exist and this document will address the physical sensor models for both scenarios. As illustrated in subsequent sections, each type of sensor has unique attributes when it comes to geopositioning. For example, the flying spot scanner will require multiple laser pulses acting as independent observations to provide area coverage and generate a 3D image, much like a whiskbroom scanner. However, some array sensors can generate a 3D range image with a single pulse obtaining a ground footprint similar to the field of view obtained in an optical image and having a similar geometric model. Other systems, although sensing with an array, require the aggregation of multiple interrogations to generate the 3D image.

2.1.2.2. Scanning / Pointing Subsystem

To generate a coverage of a target area, LIDAR systems must measure ranges to multiple locations within the area of interest. The coverage of a single instantaneous field of view (IFOV) of the system is not generally adequate to meet this need. Therefore, some combination of platform motion and system pointing is used to develop the ground coverage of a scene. This section will describe some of the pointing and scanning concepts that are being employed in current LIDAR systems.

One of the most common methods to direct the laser energy toward the ground is through a scanning mechanism. A popular scanning mechanism is an oscillating mirror which rotates about an axis through a specified angle (the angular field of view) controlling the pointing of the line of sight of the laser energy toward the ground. The mirror does not rotate completely around the axis, but oscillates back and forth by accelerating and decelerating as it scans from side to side. Oscillating mirrors are generally configured to scan perpendicular to the direction of platform motion, generating a swath width in the cross-track direction and allowing the platform motion to create coverage in the along-track direction. Oscillating mirrors create a sinusoidal scan pattern on the ground as shown in Figure 2.

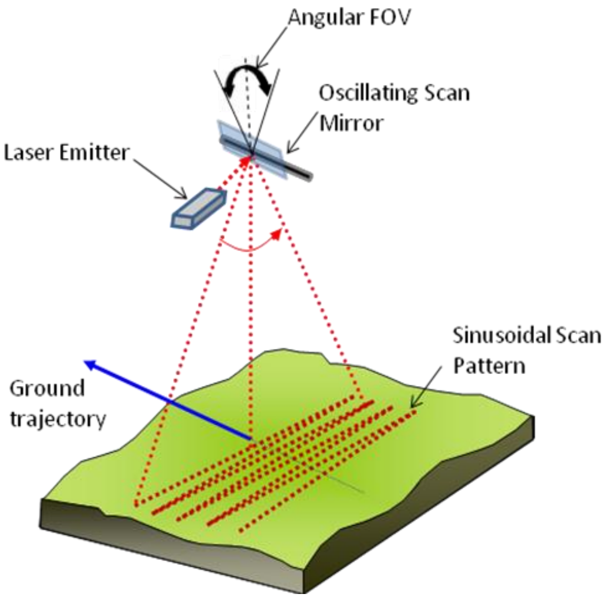


Figure 2. Oscillating Mirror Scanning System

An alternate scanning mechanism is a rotating polygon. In this system, a multifaceted polygon prism or scan mirror continuously rotates around an axis of rotation. The facets of the polygon combined with its rotation, direct the energy toward the ground. Like the oscillating system, this is generally used to sweep perpendicular to the platform trajectory generating a swath width on the ground and relying on platform motion in the along track direction to generate coverage. However, rather than relying on an oscillating

motion requiring accelerations and decelerations, the facet of the polygon controls the pointing of the continuously rotating system. As the laser energy transfers from one polygon facet to the next, there is a discontinuous and sudden jump to the opposite side of the scan resulting in a scan pattern consisting of a series of nearly parallel scan lines as shown in Figure 3. Rotating Polygon Scanning System
.

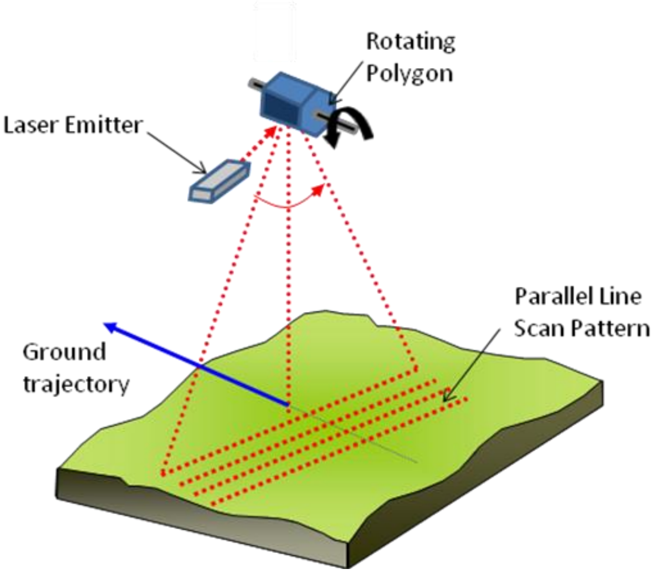


Figure 3. Rotating Polygon Scanning System

Another scanning mechanism uses a nutating mirror which is inclined in reference to the light from the laser emitter (see Figure 4. Nutating Mirror Scanning System
). The rotation of this mirror creates an elliptical scan pattern on the ground and the forward motion of the sensor creates coverage in the along track direction. (A variation on this scanning mechanism employs counter rotating Risley prisms.)

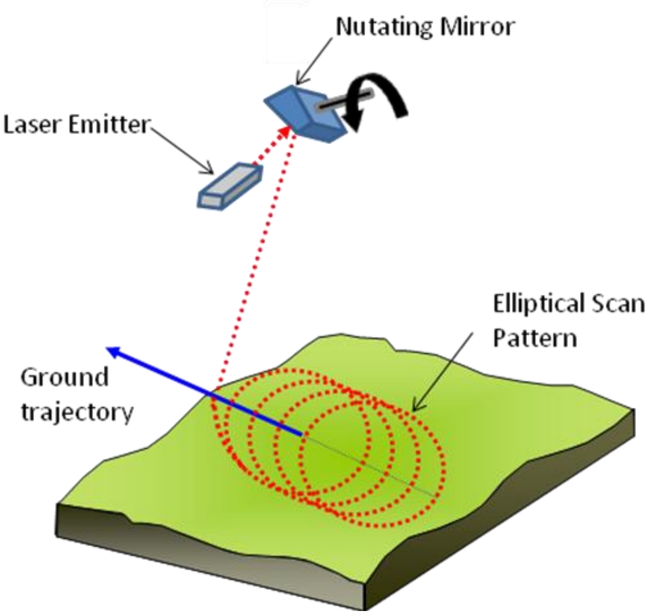


Figure 4. Nutating Mirror Scanning System

As an alternative to using a mechanical scanner, some LIDAR systems are now using fiber channels to direct the laser energy to the ground. Their goal is to achieve a more stable scan geometry due to the fixed relationship between the fiber channels and the other LIDAR components. In this system, the laser light is directed to the ground by a glass fiber bundle and the scan direction for a given pulse is dependent on which fiber channel it is emitted from. A similar system of fiber bundles are then used in the receiving optics (see Figure 5. Fiber Pointing System
).

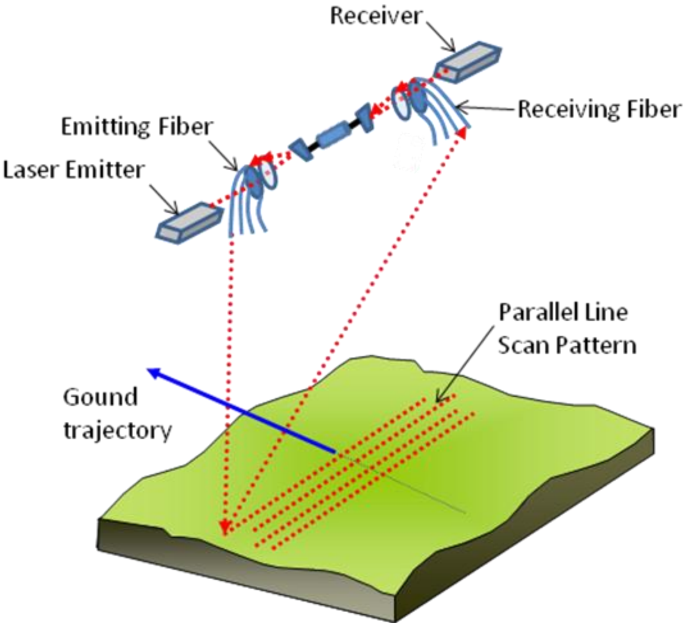


Figure 5. Fiber Pointing System

The section above illustrated several pointing methods, generally using mechanical components that are commonly used on commercial LIDAR sensors. However, the LIDAR system could also use a series of gimbals to point the line of sight. In this case, gimbals are used to rotate the line of sight around various gimbal axes. Multiple gimbal stages (that may or may not be coaxial) are used in series to obtain the desired pointing location. There are many ways that the gimbals could be driven to produce various scan patterns on the ground. The gimbals could be used exclusively to create the desired scan pattern or the gimbals could be used in conjunction with another scanning device. For example, a gimbal could be used to point the entire sensor off nadir, and another scanning system (e.g. oscillating mirror) could then be used to complete the scan pattern and control area coverage (see Figure 6. Gimbal Rotations Used in Conjunction with Oscillating Mirror Scanning System

and Figure 7. Gimbal Rotations Used to Point LIDAR System

).

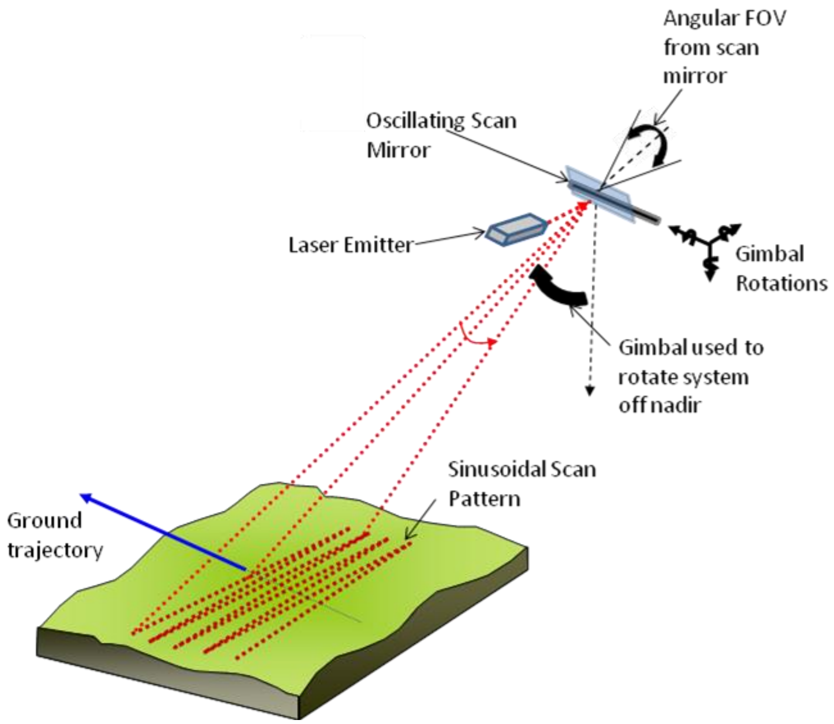


Figure 6. Gimbal Rotations Used in Conjunction with Oscillating Mirror Scanning System

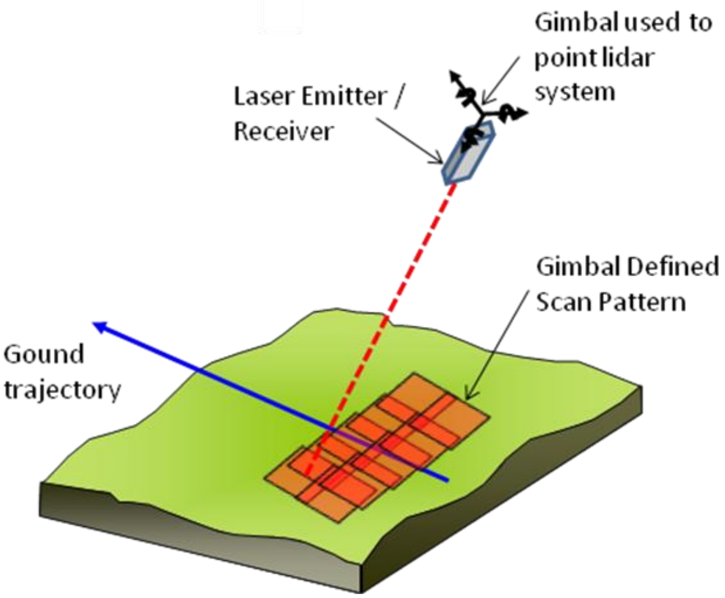

Gimbal used to point lidar system
Laser Emitter / Receiver
Gimbal Defined Scan Pattern
Gound trajectory

Figure 7. Gimbal Rotations Used to Point LIDAR System

2.1.2.3. Position and Orientation Subsystem

In the sections above, the hardware used to measure the precise ranges was described as were the techniques used to point and record data off of various locations on the ground. However, the information from these systems alone is not enough to generate a three-dimensional point cloud or range image. In addition to knowing how far away the object is (range) and the sensor pointing angles (in relationship to itself), one must also know where the platform carrying the sensor was located and how it was oriented for each incoming pulse. This information is measured and recorded by the position and orientation system.

The position and orientation system consists of two primary subsystems, the GPS and the IMU. The GPS is used to record the platform positions at a specified time interval. While there are many methods to develop GPS coordinates, the accuracies associated with LIDAR generally require a precise method such as differential post-processing with a static base station or the use of real-time differential updates. For the most accurate datasets, strict constraints are placed on the GPS base station location and on the allowable baseline separation between the GPS base station and the kinematic receiver on the platform.

The orientation of the platform is measured by an inertial measurement unit (IMU) which uses gyros and accelerometers to measure the orientation of the platform over time. Both the GPS and the IMU data are generally recorded during flight. The GPS and IMU solution will be combined (generally in a post processing step) to generate the trajectory and attitude of the platform during the data collection.

2.1.2.4. System Controller

As shown above, a LIDAR system consists of many sub-components that have to work together to generate a dataset. The quality and density of the output product is dependent on the operation and settings of the subsystems. As the name implies, the system controller is used to provide the user an interface to the system components and coordinate their operation. It allows the operator to specify sensor settings and to monitor the operation of the subsystems.

2.1.2.5. Data Storage

Raw LIDAR data includes files from the GPS, the IMU, the ranging unit, and possibly other system components. Even in its raw state, LIDAR systems can generate massive quantities of data. Due to the quantities of data, the datasets are often stored with the system and downloaded after collection. The Data Storage unit is used to store the data from all of the system components.

2.2 LIDAR Data Processing Levels

Several processing steps are necessary to create a useable "end product" from raw LIDAR data. However, the resultant form of the data at intermediate processing steps may be of use to different groups within the community. In order to classify the degree of LIDAR processing applied to a given dataset, the LIDAR community has initiated defining multiple LIDAR data processing levels. Each level describes the *processing state* of the data. Following are definitions of the levels (denoted L0 through L5), along with basic descriptions of the processing involved between levels and the types of users each level would apply to.

2.2.1. Level 0 (L0) – Raw Data and Metadata

L0 data consists of the raw data in the form it is stored in as collected from the mapping platform. The dataset includes, but is not limited to, data from GPS, IMU, laser measurements (timing, angles) and gimbal(s). Metadata would include items such as the sensor type, date, calibration data, coordinate frame, units and geographic extents of the collection. Other ancillary data would also be included, such as GPS observations from nearby base stations. Typical users of L0 data would include sensor builders and data providers, and also researchers looking into improving the processing of L0 to L1 data.

2.2.2. Level 1 (L1) – Unfiltered 3D Point Cloud

L1 data consists of a 3D point data (point cloud) representation of the objects measured by the LIDAR mapping system. It is the result of applying algorithms (from sensor models, Kalman filters, etc.) in order to project the L0 data into 3-space. All metadata necessary for further processing is also carried forward at this level. Users would include scientists and others working on algorithms for deriving higher-level datasets, such as filtering or registration.

2.2.3. Level 2 (L2) – Noise-filtered 3D Point Cloud

L2 data differs from L1 in that noisy, spurious data has been removed (filtered) from the dataset, intensity values have been determined for each 3D point (if applicable) and relative registration (among scans, stares or swaths) has been performed. The impetus behind separating L1 from L2 is due to the nature of Geiger-mode LIDAR (GML) data. Derivation of L1 GML data produces very noisy point clouds, which requires specialized processing (*coincidence processing*) to remove the noise. Coincidence processing algorithms are still in their infancy, so their ongoing development necessitates a natural break in the processing levels. As with L1, all metadata necessary for further processing is carried forward. Typical users include exploitation algorithm developers and scientists developing georegistration techniques.

2.2.4. Level 3 (L3) – Georegistered 3D Point Cloud

L3 datasets differ from L2 in that the data has been registered to a known geodetic datum. This may be performed by an adjustment using data-identifiable objects of known geodetic coordinates or some other method of control extension for improving the absolute accuracy of the dataset. The primary users of L3 data would be exploitation algorithm developers.

2.2.5. Level 4 (L4) – Derived Products

L4 datasets represent LIDAR-derived products to be disseminated to standard users. These products could include Digital Elevation Models (DEMs), viewsheds or other products created in a standard format and using a standard set of tools. These datasets are derived from L1, L2 or L3 data, and are used by the basic user.

2.2.6. Level 5 (L5) – Intel Products

L5 datasets are a type of specialized products for users in the intelligence community, which may require specialized tools and knowledge to generate. The datasets are derived from L1, L2 or L3 data.

3. Coordinate Systems

A sensor model uses measurements from various system components to obtain geographic coordinates of the sensed objects. However, the system components are not centered nor aligned with a geographic coordinate system. The reference frame of each component and their interrelationships must be understood to obtain geographic coordinates of a sensed object. The following sections will define these coordinate systems and their interrelationships.

3.1 General Coordinate Reference System Considerations

The purpose of a sensor model is to develop a mathematical relationship between the position of an object on the Earth's surface and its data as recorded by a sensor. The spatial positions of the sensor during data collection may be given, at least initially or in its raw form, either in relation to a coordinate system locally defined or relative to an Earth reference. A 3-dimensional datum will be required to define the origin and orientation of the coordinate systems. Likewise, the positions of the objects may be defined with respect to either the same coordinate system, or attached to any number of Earth-based datums (e.g. WGS-84). For purposes of this metadata profile, the transformations among the various coordinate systems will be accomplished via a sequence of translations and rotations of the sensor's coordinate system origin and axes until it coincides with an Earth-based coordinate system origin and axes. An overall view of some of the coordinate reference systems under consideration is shown in **Error! Reference source not found.**.

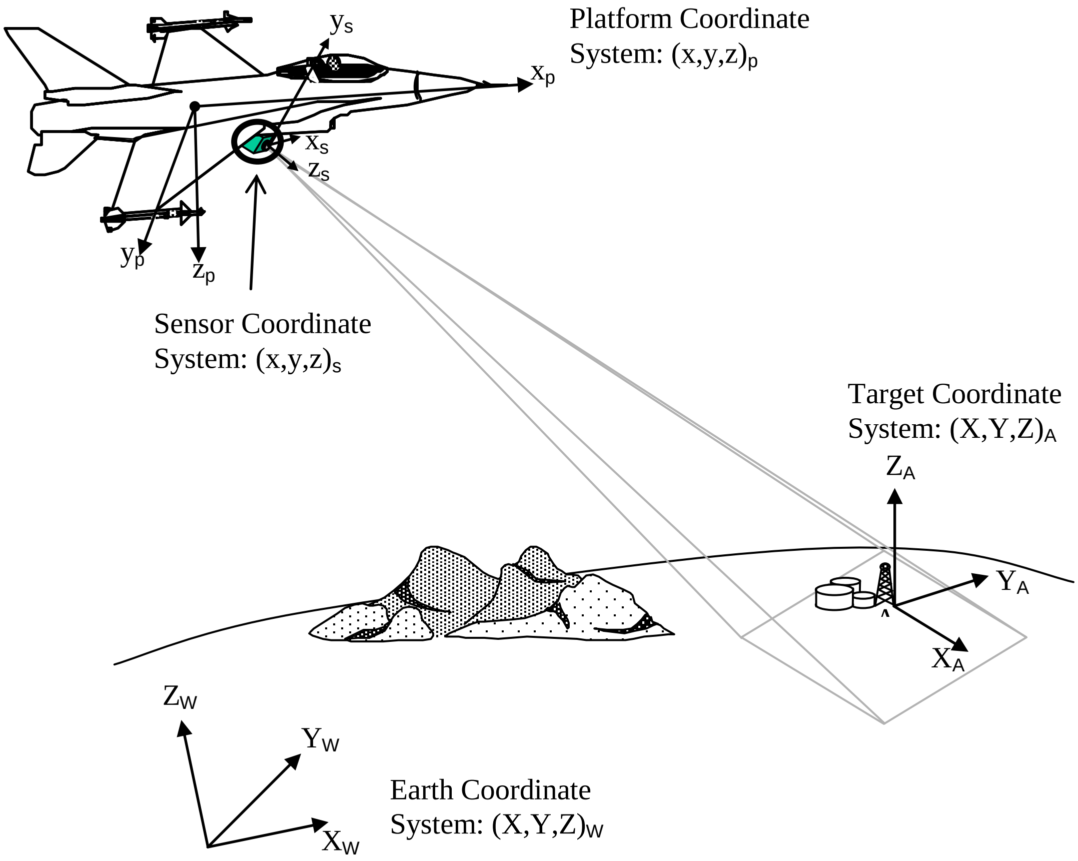

y_s
Platform Coordinate System: $(x,y,z)_p$
x_p
x_s
z_s
y_p
z_p
Sensor Coordinate System: $(x,y,z)_s$
Target Coordinate System: $(X,Y,Z)_A$
Z_A
Y_A
X_A
Z_W
Y_W
X_W
Earth Coordinate System: $(X,Y,Z)_W$

Figure 8. Multiple coordinate reference systems

The sensor position may be described in many ways and relative to any number of coordinate systems particularly those of the aerial platform. There may also be one or more gimbals to which the sensor is attached, each with its own coordinate system, in addition to the platform's positional reference to the Global Positioning System (GPS) or other onboard inertial measurement unit (IMU). Transformations among coordinate systems can be incorporated into the mathematical model of the sensor.

Airborne platforms normally employ GPS and IMU systems to define position and attitude. The GPS antenna and the IMU gyros and accelerometers typically are not physically embedded with the sensor. For a GPS receiver the point to which all observations refer is the phase center of the antenna. The analogous point for an IMU is the intersection of the three sensitivity axes. The physical offset between the two generally is termed a lever arm. Denoting the lever arm vector from the GPS antenna phase center to the IMU is the vector $\boldsymbol{r}_{GPS}$. An analogous lever arm between the IMU and the sensor is labeled $\boldsymbol{r}_{IMU}$. These relationships are illustrated in Figure .

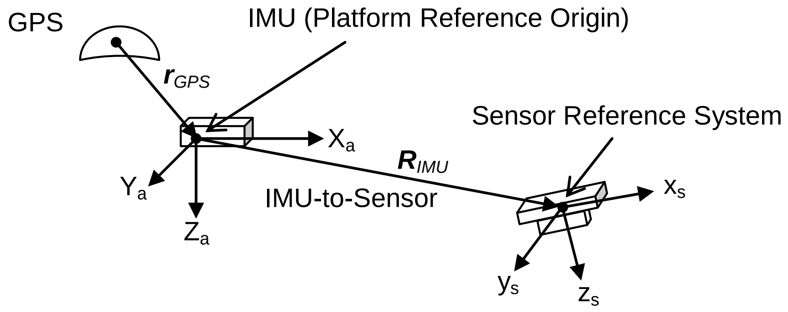


Figure 9. Nominal Relative GPS to IMU to Sensor Relationship

3.2 Scanner Coordinate Reference System

This system describes the reference frame of the scanner during a laser pulse firing. The origin of this system is the location where the laser emerges from the lidar system toward the ground. The system axes are defined as follows: z-axis (z_{sc}) positive is aligned with the laser pulse vector; with scan angles set to zero, x-axis (x_{sc}) positive is aligned with the Sensor Reference System x-axis, described below; y-axis (y_{sc}) positive is chosen to complete a right-handed Cartesian system. Non-zero scan angles will cause the x-axis and/or the y-axis to deviate from alignment with the sensor reference system.

3.3 Sensor Coordinate Reference System

This system describes the reference frame of the sensor, in which the scanner operates. The scanner reference system rotates within this system as the scan angles change, and is coincident with this system when scan angles are zero. The origin of this system is located at the origin of the Scanner Coordinate Reference System. The system axes (Figure 9) are defined as follows: z-axis (z_s) positive is nominally aligned with nadir, although this could depend on the mount configuration; x-axis (x_s) positive is referenced to a chosen direction in the scanner plane (orthogonal to the z-axis) which is nominally aligned with the flight direction when no z-rotation is applied to the gimbals; y-axis (y_s) positive is chosen to complete a right-handed Cartesian system.

3.4 Gimbal Coordinate Reference System

This system describes the reference frame of a gimbal, which houses the sensor and orients it depending on applied gimbal angles. The origin of this system is located at the intersection of the gimbal axes. With gimbal angles set to zero, the gimbal axes are defined as follows: x-axis (x_{GIM}) positive is nominally aligned with the flight direction; y-axis (y_{GIM}) positive is orthogonal to the x-axis and points out the right side of the aircraft; z-axis (z_{GIM}) positive points downward, completing a right-handed Cartesian system. Multiple gimbals or gimbal stages may be used in a system, and the axes may not be coaxial. Therefore multiple gimbal reference systems may be defined.

3.5 Platform Coordinate Reference System

This system describes the reference frame of the aircraft platform, to which the gimbal is mounted. The origin is located at the aircraft center of navigation (i.e. the IMU center of rotation). The axes are defined as follows: x-axis (x_p) positive along the heading of the platform, along the platform roll axis; y-axis (y_p) positive in the direction of the right wing, along the pitch axis; z-axis (z_p) positive down, along the yaw axis. Any rotational differences between the gimbal reference system and the platform reference system describe the rotational boresight (or mounting) angles, which are fixed for a given system installation.

3.6 Local-vertical Coordinate Reference System

This system describes the reference frame with respect to the local-vertical. Coordinates in this system are obtained by applying IMU measurements to coordinates in the Platform Reference System. The origin is located at the aircraft center of navigation (i.e. the IMU center of rotation). The axes are defined as follows: z-axis (z_a) positive points downward along the local gravity normal; x-axis (x_a) positive points toward geodetic north; y-axis (y_a) positive points east, completing a right-handed Cartesian system.

The platform reference system is related to the local-vertical reference system with its origin at the center of navigation. In horizontal flight, the platform z-axis is aligned with the local gravity normal. The platform reference system orientation stated in terms of its physical relationships (rotations) relative to this local-vertical reference (**Error! Reference source not found.**) is as follows:

- Platform heading - horizontal angle from north to the platform system x-axis X_a (positive from north to east).
- Platform pitch - angle from the local-vertical system horizontal plane to the platform positive x-axis X_a (positive when positive x-axis is above the local-vertical system horizontal plane or nose up).
- Platform roll - rotation angle about the platform x-axis; positive if the platform positive y-axis Y_a lies below the local-vertical system horizontal plane (right wing down).

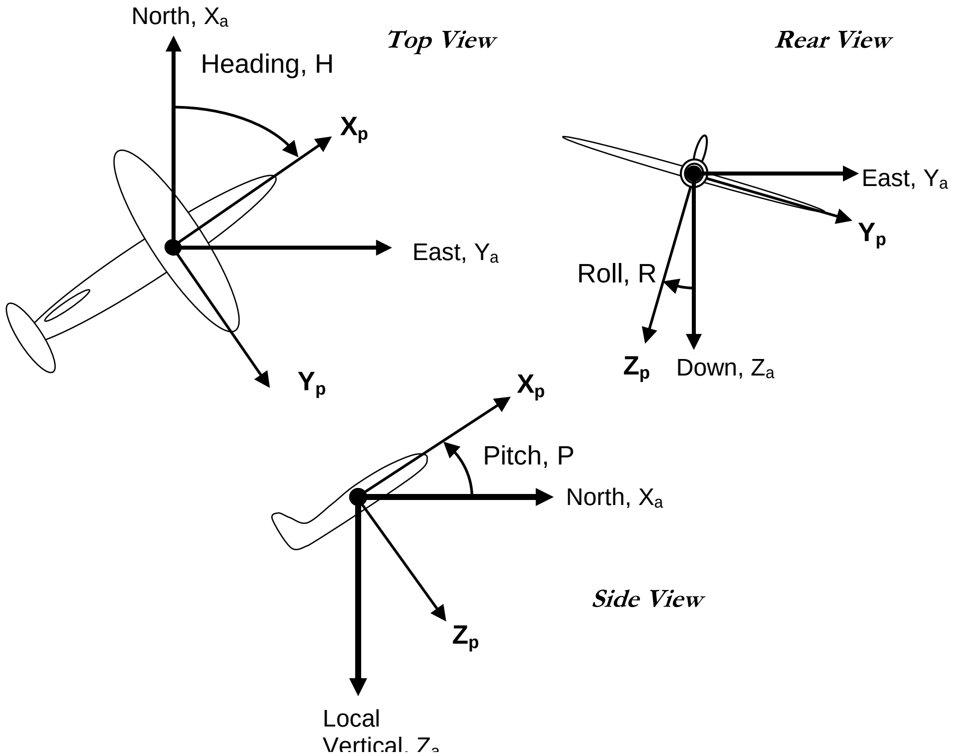


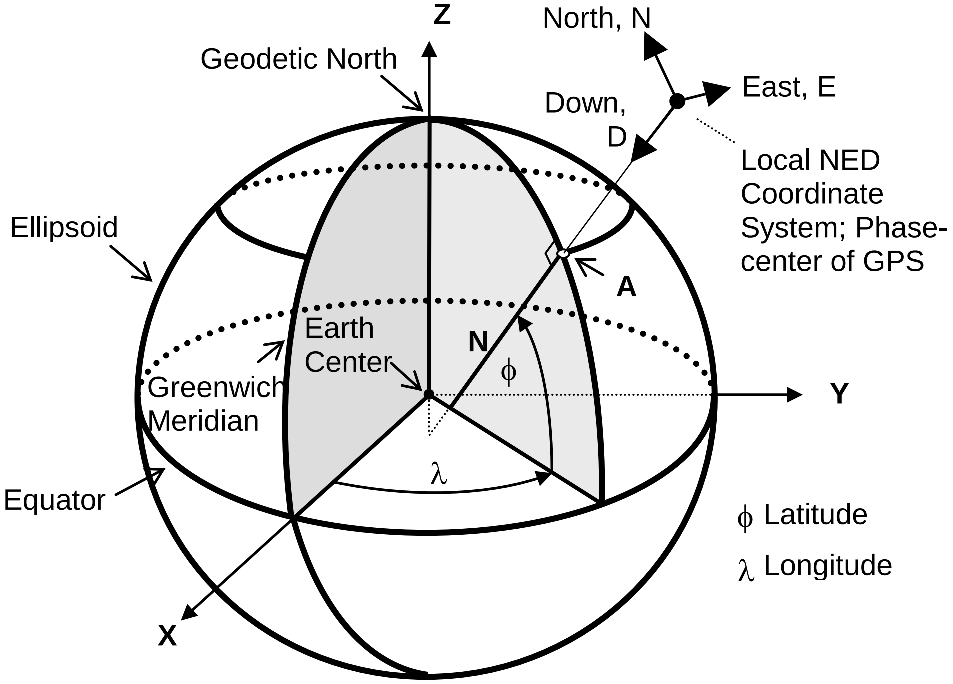
Figure 10. Relationship between the platform reference system (XpYpZp) and local-vertical system

3.7 Ellipsoid-tangential (NED) Coordinate Reference System

This system describes the North-East-Down (NED) reference frame with the horizontal plane tangent to the geodetic ellipsoid to be referenced (i.e. WGS-84). The difference between this system and the local-vertical system is the angular difference between the ellipsoid normal and the local gravity normal. This angle between the normals (also the angle between the z-axes of the two coordinate systems) is known as the *deflection of the vertical*. The origin of the NED system is located at the phase-center of the GPS antenna, fixed to the platform structure. The axes are defined as follows: z-axis positive points downward along the ellipsoidal normal; x-axis positive points toward geodetic north; y-axis positive points east, completing a right-handed Cartesian system.

3.8 ECEF Coordinate Reference System

This system describes the Earth-Centered Earth-Fixed (ECEF) reference frame of the geodetic ellipsoid to be referenced (i.e. WGS-84). GPS measurements reference this system. The origin is located at the origin of the geodetic ellipsoid, which is the geocenter or center of mass of the earth. The axes are defined as follows: z-axis positive points along Earth's rotational axis toward geodetic north; x-axis positive points toward the 0-degree longitudinal meridian; y-axis positive completes a right-handed Cartesian system. The relationship between the NED reference system and the ECEF reference system is illustrated in Figure 11. ECEF and NED coordinate systems



.

Figure 11. ECEF and NED coordinate systems

Any point may be described in geocentric (X,Y,Z) coordinates, or alternatively in the equivalent geodetic latitude, longitude and ellipsoid height terms. Also, a point can be described relative to a local reference system with origin on an Earth-related ellipsoidal datum (e.g. WGS-84 ellipsoid), specifically in an East-North-Up (ENU) orientation; where North is tangent to the local prime meridian and points North, Up points upward along the ellipsoidal normal, and East completes a right-hand Cartesian coordinate system.

shows an ENU system and its relationship to an ECEF system.

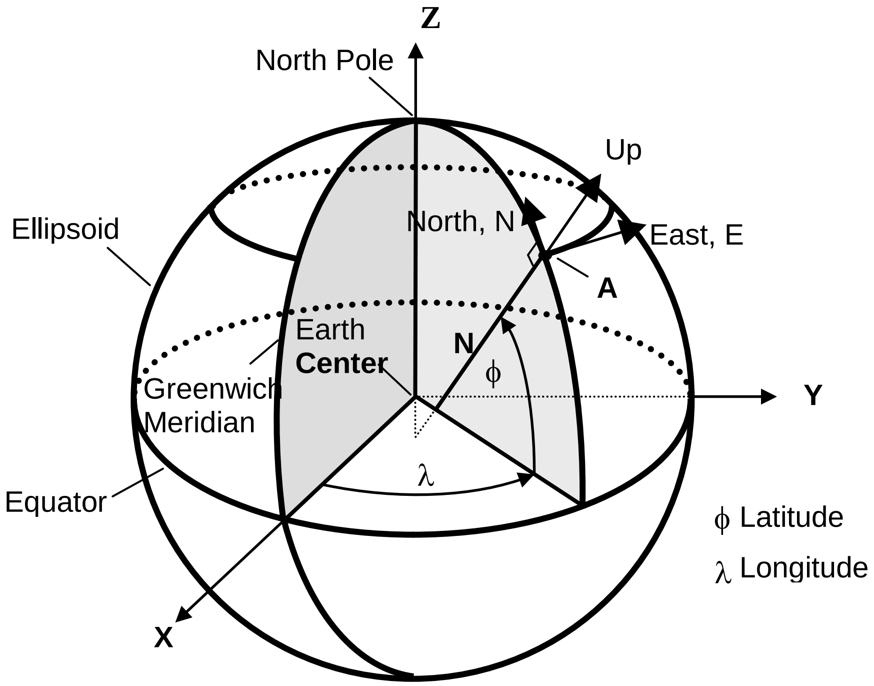


Figure 12. Earth-centered (ECEF) and local surface (ENU) coordinate systems (MIL-STD-2500C)

4. Sensor Equations

This section outlines the equations representing the spatial relationships among the various components of a LIDAR collection system. Equations particular to point-scanning systems are described first, followed by equations particular to frame-scanning systems.

4.1 Point-scanning Systems

The relationships among some basic components of a LIDAR collection system are illustrated in **Error! Reference source not found.**, including the GPS, IMU and sensor. The phase-center of the GPS antenna provides the connection to the ECEF reference datum (e.g. WGS-84). A series of translations and rotations, obtained from sensor observations and constants, must be applied to a LIDAR pulse measurement for direct geopositioning of the sensed ground object.

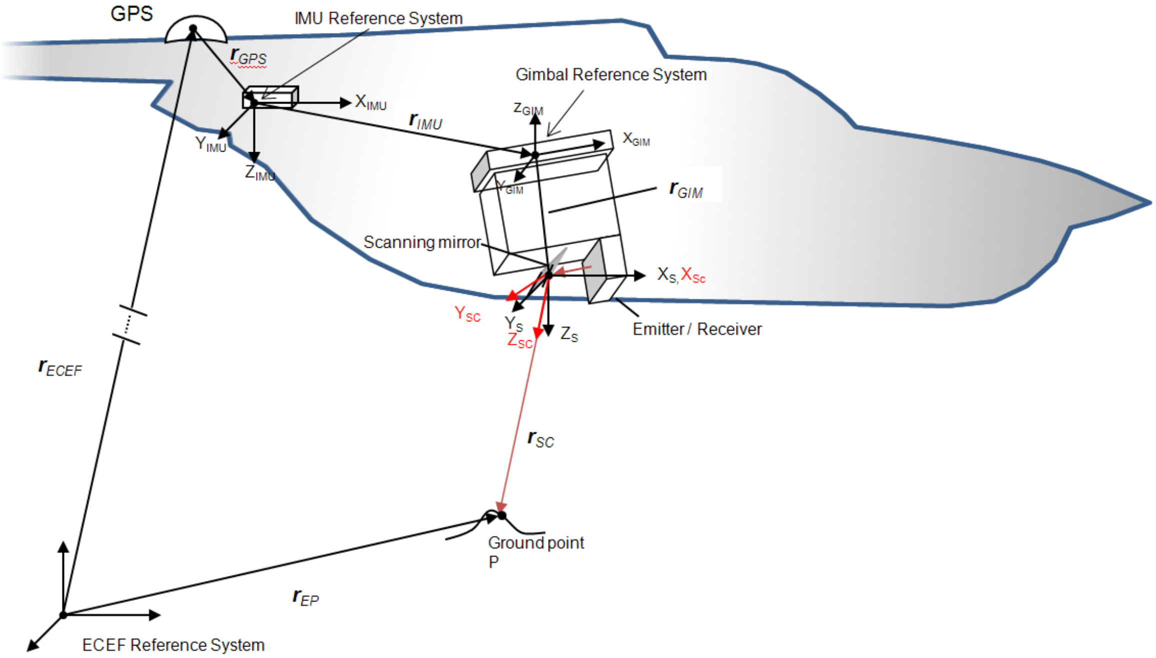


Figure 13: Lidar coordinate systems and their relationship

The coordinates of a sensed ground point in a geocentric ECEF coordinate system (e.g. WGS-84) are obtained from the following equation:

$$r_{EP} = r_{ECEF} + M_{ECEF} M_{ELL} M_{VER} (M_{PLA} M_{GIM} \llbracket(M\rrbracket_{SEN} r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS}) \qquad \textbf{Eq. 3}$$

The components of $r_{EP} = r_{ECEF} + M_{ECEF} M_{ELL} M_{VER} (M_{PLA} M_{GIM} \llbracket(M\rrbracket_{SEN} r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3 are described below:

r_{SCA} vector from the scanner to the ground point in the scanner reference frame (range)
r_{GIM} vector from the gimbal center of rotation to the sensor in the gimbal reference frame
r_{INSMU} vector from the IMU to the gimbal center of rotation in the platform reference frame
r_{GPS} vector from the GPS antenna phase-center to the IMU in the platform reference frame
r_{ECEF} vector from the ECEF origin to the GPS antenna phase-center in the ECEF reference frame (GPS observations)
r_{EP} vector from the ECEF origin to the ground point in the ECEF reference frame
M_{SEN} rotation matrix from scanner reference frame to sensor reference frame (scan angles)
M_{GIM} rotation matrix from the sensor reference frame to the gimbal reference frame (gimbal angles)
M_{PLA} rotation matrix from the gimbal reference frame to the platform reference frame (boresight angles)
M_{VER} rotation matrix from the platform reference frame to the local-vertical reference frame (IMU observations)
M_{ELL} rotation matrix from the local-vertical reference frame to the ellipsoid-tangential reference frame
M_{ECEF} rotation matrix from the ellipsoid-tangential (NED) reference frame to the ECEF reference frame

The components r_{INS}, r_{GPS} and M_{PLAGIM} are constants which are measured at system installation or determined by system calibration. Appendix C: Coordinate System Transformations provides a general introduction into the development of coordinate system transformations.

Note that the vector r_{SCA} does not account for any internal laser propagation within the system, both before the laser is emitted or after it is detected. It is assumed that any such offsets are accounted for by the hardware or processing software in order to provide a measurement strictly from the scanner to the ground point.

Other system component configurations are possible which would alter $r_{EP} = r_{ECEF} + M_{ECEF}M_{ELL}M_{VER}(M_{PLA}M_{GIM}\llbracket(M\rrbracket_{SEN}r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3. Some systems have the IMU mounted on the back of the sensor, which would cause r_{GPS} to vary with the gimbal settings. In this case, the *distance* from the IMU to the gimbal rotational center would be constant, and a vector (constant) from the GPS antenna to the gimbal rotational center would be needed.

4.1.1. Atmospheric Refraction

Light rays passing through media with differing refractive indices are refracted according to Snell's Law. This principle applies to laser beams passing downward through the atmosphere, as the refractive index of the atmosphere changes with altitude. The effect is an angular displacement of the laser beam as described in Eq. 4 below:

$$\Delta d = K \tan \alpha \qquad \textbf{Eq. 4}$$

Δd angular displacement of the laser beam from the expected path
α the angle of the laser beam from vertical
K a constant, defined below

Several models are available to determine the constant *K*, however a commonly used model developed by the Air Force is the Air Research and Development Command (ARDC) model. Using this model, the constant *K* is determined as follows:

$$K = \frac{2410H}{H^2 - 6H + 250} - \frac{2410h}{h^2 - 6h + 250}\left(\frac{h}{H}\right) \qquad \textbf{Eq. 5}$$

where

H flying height (MSL) of the aircraft, in kilometers
h height (MSL) of the object the laser intersects, in kilometers

Applying *H* and *h* in kilometers, the resulting units for the constant *K* are microradians.

Since the angle *α* is relative to vertical, it can be derived from Eq. 3 using a chain of rotation matrices ($M_{VER}M_{PLA}M_{GIM}M_{SEN}$). Then the calculation of Δ*d* is applied to M_{VER} resulting in a new value, M'_{VER}, which is substituted into Eq. 3.

The above equations are appropriate for most mapping scenarios, however at very large oblique vertical angles (> 60°) a spherically stratified model should be applied (Gyer, 1996). Snell's law for a spherically stratified model is represented by the following:

$$n_s h_s \sin(\alpha_s) = n_g h_g \sin(\alpha_g) = k = constant \qquad \textbf{Eq. 6}$$

where

n_s , n_g index of refraction at sensor and ground point, respectively
h_s , h_g ellipsoid height of sensor and ground point, respectively
α_s , α_g the angle of the laser beam from vertical at the sensor and ground point, respectively

The angular displacement Δd is obtained from the following equation:

$$\tan\Delta d = \frac{\tan\alpha\left(\left({}^{r_s}/_{r_g}\right) - \cos\theta\right) - \sin\theta}{\tan\alpha\,\sin\theta + \left({}^{r_s}/_{r_g}\right) - \cos\theta}$$

Eq. 7

where

Δd angular displacement of the laser beam from the expected path
α angle of the laser beam from vertical
r_s height of the scanner above center of sphere (approximating ellipsoid curvature)
r_g height of the illuminated ground point above center of sphere
θ angle subtended at the center of sphere, from the scanner to the ground point

The value for θ is determined from the integral

$$\theta = \int_{h_g}^{h_s} \frac{k}{r\sqrt{n^2 r^2 - k^2}}\,dh$$

Eq. 8

where h_s and h_g are the ellipsoid heights at the sensor and ground point, respectively. Rather than using the ellipsoid directly, Gyer uses a sphere to approximate the local ellipsoid curvature, and θ is the angle between two vectors within this sphere: center of sphere to the ground point, and center of sphere to the scanner. The value of θ can be estimated using numerical integration (see Gyer, 1996). The value for n can be computed from

$$n \approx 1 + 0.000078831\,\frac{P}{T}$$

Eq. 9

where T (temperature) and P (pressure) are in degrees Kelvin and millibars, respectively. Lastly, if local measurements are not available, the values of T and P can be calculated from the following:

$$T = 288 - 0.00649h$$

Eq. 10

$$P = 1013\left[{}^{T}/_{288}\right]^{5.256}$$

Eq. 11

Again T is in Kelvin, P is in millibars and h (altitude) is in meters above MSL.

If adjustments to n for wavelength are needed, the following equations can be applied (Anderson and Mikhail, 1998) which were modified for consistent units

$$n_g = 1 + \left(287.604 + \frac{4.8864}{\lambda^2} + \frac{0.068}{\lambda^4}\right)10^{-6}$$

Eq. 12

$$n = 1 + \frac{0.2702812(n_g - 1)P}{T} - \frac{1.12977e(10^{-5})}{T} \qquad \textbf{Eq. 13}$$

where λ is wavelength in micrometers, e is vapor pressure in millibars and n_g is the index of refraction of standard air for the given wavelength λ.

4.2 Frame-scanning Systems

Frame-scanning LIDAR systems use the same basic system components as point-scanning systems (Figure 1); however the receiver consists of an array of detector elements (similar to an imaging system) rather than a single detector. This differing receiver geometry is described by its own coordinate system and has inherent geometric and optical effects which must be accounted for. Following is a description of the frame coordinate system, the corrections necessary for a frame system, and the resulting sensor modeling equations. Much of the information in this section was obtained from the Frame Camera Formulation Paper.

4.2.1. Frame Coordinate System

A frame sensor is a digital collection array consisting of a matrix of detectors, or elements, at the focal plane (**Error! Reference source not found.**). The Focal Plane Array (FPA) origin is located at the intersection of the sensor optical axis and image plane. Since reference is made to a positive image, the focal plane and sensor axes will be aligned.

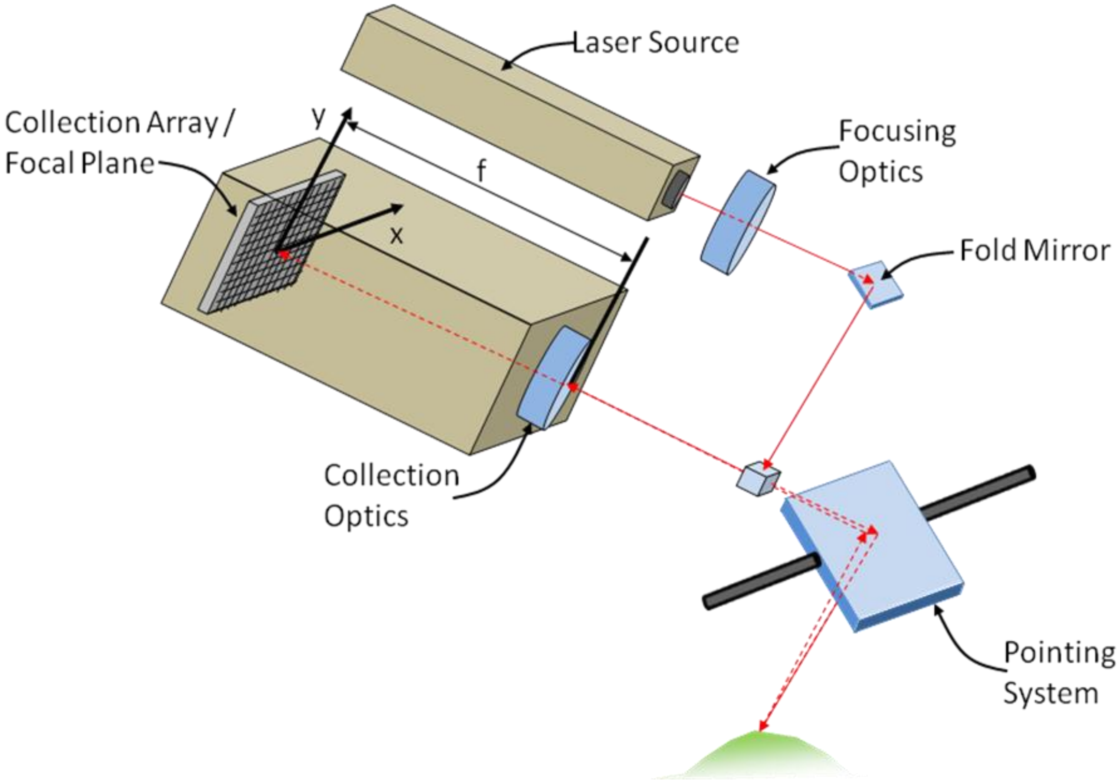


Figure 14. Receiver / Emitter system in lidar

Typical of common imagery formats, and in particular ISO/IEC 12087-5, pixels are indexed according to placement within a “Common Coordinate System” (CCS), a two-dimensional array of rows and columns, as illustrated in the array examples in **Error! Reference source not found.**. There are three commonly referred to coordinate systems associated with digital and digitized imagery: row, column (r,c), line, sample (ℓ,s), and x,y. The units used in the first two systems are pixels, while the x,y are linear measures such as millimeters. The origin of the CCS (and the line/sample system), as shown in **Error! Reference source not found.**, is the upper left corner of the first (or 0,0) pixel, which in turn is the upper left of the array. Because the CCS origin is the pixel corner, and the row/column associated with a designated pixel refers to its center, the coordinates of the various pixels, (0.5,0.5), (0.5,1.5), etc., are not integers.

4.2.1.1. Row-Column to Line-Sample Coordinate Transformation

Typical frame processing is based on the geometric center of the image as the origin. As shown in **Error! Reference source not found.**, the origin of the row-column system is located in the upper-left corner of the array, and the origin of the line-sample system is in the center of the array. The positive axes of the systems are parallel and point in the same direction, so conversion between the two systems is attained by applying simple translation offsets:

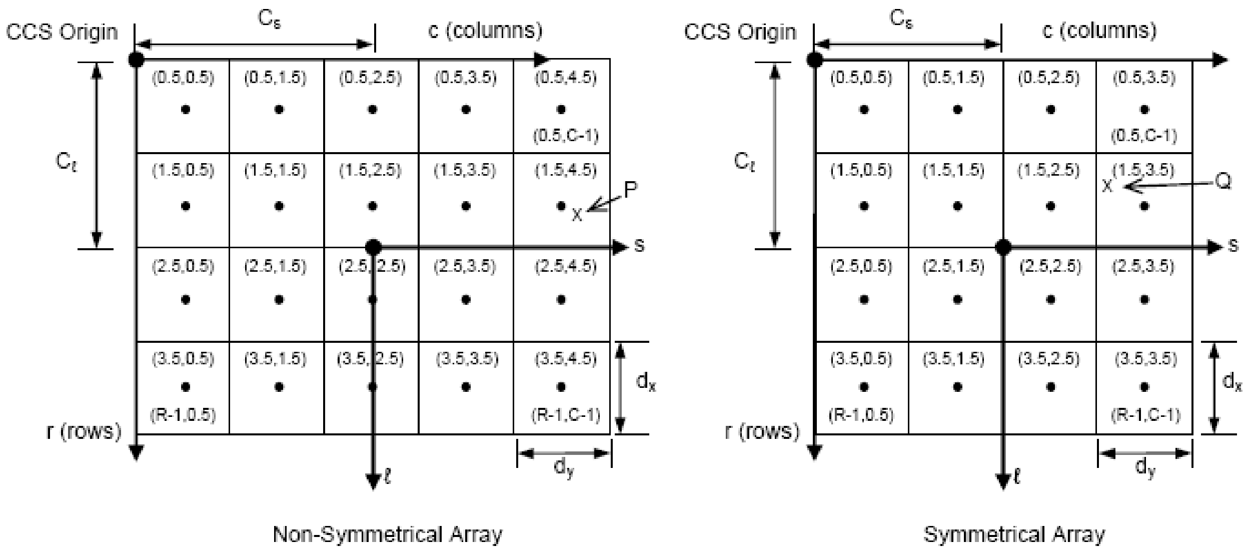
$$\ell = r - C_\ell$$

Eq. 14

$$s = c - C_s$$

Eq. 15

where C_ℓ and C_s are each half the array size, in pixels, in the row and column directions, respectively.



Non-Symmetrical Array

Symmetrical Array

Figure 15. Coordinate systems for non-symmetrical and symmetrical arrays

4.2.2. Frame Corrections

Corrections to the interior of the frame system, including array distortions, principal point offsets and lens distortions, and exterior corrections such as atmospheric refraction, are described in the following sections.

4.2.2.1. Array Distortions

Distortions in the array are accounted for by the following equations:

$$x = a_1\ell + b_1 s + c_1$$

Eq. 16

$$y = a_2\ell + b_2 s + c_2$$

Eq. 17

This transformation accounts for two scales, a rotation, skew, and two translations. The resulting *x* and *y* values are typically in millimeter units. The six parameters (*a₁, b₁, c₁, a₂, b₂, c₂*) are usually estimated on the basis of (calibrated) reference points, such as corner pixels for digital arrays. The (x,y) image coordinate system, as shown in Figure 16. (x,y) Image Coordinate System and Principal Point Offsets

, is used in the further construction of the mathematical model.

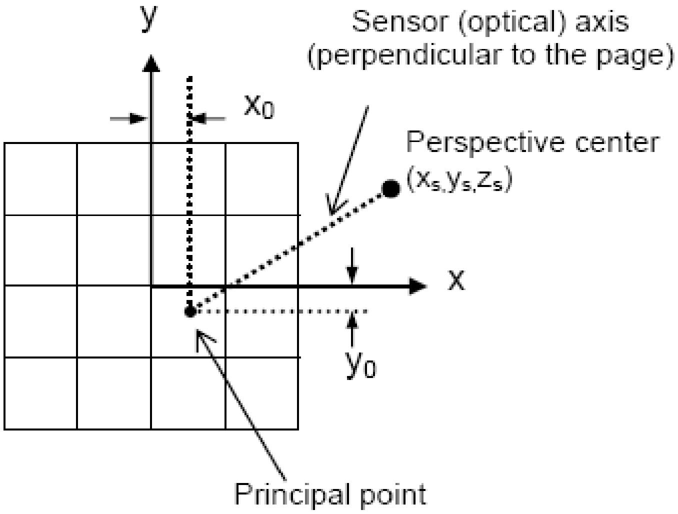


Figure 16. (x,y) Image Coordinate System and Principal Point Offsets

4.2.2.2. Principal Point Offsets

Ideally the sensor (lens) axis would intersect the collection array at its center coordinates (x=0,y=0). However, this is not always the case due to lens flaws, imperfections, or design, and is accounted for by offsets x_0 and y_0, as shown in Figure 16. (x,y) Image Coordinate System and Principal Point Offsets

. Note that x_0 and y_0 are in the same linear measure (e.g., mm) as the image coordinates (x,y) and the focal length, *f*. For most practical situations, the offsets are very small, and as such there will be no attempt made to account for any covariance considerations for these offset terms.

4.2.2.3. Lens Distortions

Radial lens distortion is the radial displacement of an imaged point from its expected position (Mikhail et al.). Figure 17. Radial Lens Distortion image coordinate components

illustrates this distortion and its *x* and *y* image coordinate components. Calibration procedures are employed to determine radial lens distortion, and it is typically modeled as a polynomial function of the radial distance from the principal point, as provided below:

$$\delta r = k_1 r^3 + k_2 r^5 + k_3 r^7 \qquad \textbf{Eq. 18}$$

where

$$r = \sqrt{\bar{x}^2 + \bar{y}^2} \qquad \bar{x} = x - x_0 \qquad \bar{y} = y - y_0 \qquad \textbf{Eq. 19}$$

and k_1, k_2 and k_3 are radial lens distortion parameters derived from calibration.

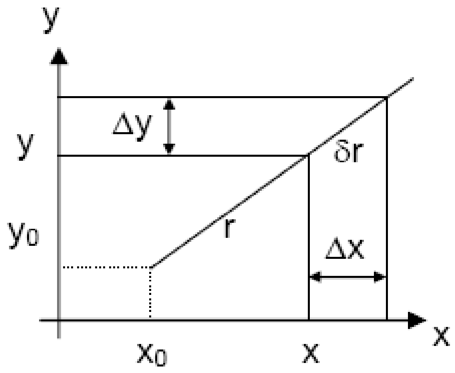


Figure 17. Radial Lens Distortion image coordinate components

The effect of radial lens distortion on the *x* and *y* image coordinate components is:

$$\Delta x_{radial} = \bar{x}\frac{\delta r}{r} = \bar{x}\frac{k_1 r^3 + k_2 r^5 + k_3 r^7}{r} = \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6) \qquad \textbf{Eq. 20}$$

$$\Delta y_{radial} = \bar{y}\frac{\delta r}{r} = \bar{y}\frac{k_1 r^3 + k_2 r^5 + k_3 r^7}{r} = \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6) \qquad \textbf{Eq. 21}$$

Another lens correction is decentering (or tangential lens distortion), which is caused by errors in the assembly of the lens components and affects its rotational symmetry (Mikhail et al.). This correction is typically insignificant, although it can be more prominent in variable focus or zoom lenses. The *x* and *y* image coordinate components of decentering are commonly modeled by the following equations:

$$\Delta x_{dec} = p_1(2\bar{x}^2 + r^2) + p_2(2\bar{x}\bar{y})$$ **Eq. 22**

$$\Delta y_{dec} = p_1(2\bar{x}\bar{y}) + p_2(2\bar{y}^2 + r^2)$$ **Eq. 23**

where p1 and p2 are decentering coefficients derived from calibration. Combining lens corrections to image coordinates from radial lens distortion (Eq. 20, Eq. 21) and decentering (Eq. 22, Eq. 23) results in the following:

$$\Delta x_{lens} = \Delta x_{radial} + \Delta x_{dec} = \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(2\bar{x}^2 + r^2) + p_2(2\bar{x}$$

Eq. 24

$$\Delta y_{lens} = \Delta y_{radial} + \Delta y_{dec} = \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(2\bar{x}\bar{y}) + p_2(2\bar{y}^2 + r$$

Eq. 25

It should be noted that for some lidar systems, lens distortions are not modeled using the typical photogrammetric distortion parameters presented here. Instead, distortions in the combination of the detector and optical path are determined and accounted for during lab testing. Rather than carry distortion parameters for the entire system, a calibrated line-of-sight (LOS) vector associated with each pixel in a detector is determined and carried with the dataset. This unique LOS is then used during the ray trace to determine the final ground coordinates of a point measured from a specified pixel.

4.2.2.4. Atmospheric Refraction

The principle of atmospheric refraction for a frame-scanning system is the same as that given by equations Eq. 4 and $K = \dfrac{2410H}{H^2 - 6H + 250} - \dfrac{2410h}{h^2 - 6h + 250}\left(\dfrac{h}{H}\right)$ Eq. 5 for the point-scanning system. However, the frame receiver geometry causes the application of the correction to be similar to that for radial lens distortion. Given equations Eq. 4 and $K = \dfrac{2410H}{H^2 - 6H + 250} - \dfrac{2410h}{h^2 - 6h + 250}\left(\dfrac{h}{H}\right)$ Eq. 5, the corrected *x* and *y* image coordinates are shown below:

$$x'_{atm} = \frac{\bar{x}(r'_{atm})}{r}$$ **Eq. 26**

$$y'_{atm} = \frac{\bar{y}(r'_{atm})}{r}$$ **Eq. 27**

where

$$r = \sqrt{\bar{x}^2 + \bar{y}^2} \qquad \bar{x} = x - x_0 \qquad \bar{y} = y - y_0$$

and

$$r'_{atm} = f\tan(\alpha + \Delta d)$$ **Eq. 28**

Therefore the image coordinate corrections are:

$$\Delta x_{atm} = x'_{atm} - \bar{x} = \bar{x}\left(\frac{r'_{atm}}{r} - 1\right)$$

Eq. 29

$$\Delta y_{atm} = y'_{atm} - \bar{y} = \bar{y}\left(\frac{r'_{atm}}{r} - 1\right)$$

Eq. 30

A spherically stratified model is needed for highly oblique (> 60°) vertical angles. For this formulation, first the image coordinates of the nadir point are calculated using:

$$x_n = -f\frac{m_{13}}{m_{33}}$$

Eq. 31

$$y_n = -f\frac{m_{23}}{m_{33}}$$

Eq. 32

where m_{13}, m_{23} and m_{33} are the rotation matrix components (Eq. 50) from the sensor to ECEF reference frames. The distance from the image nadir coordinates to the imaged object coordinates $\bar{x}$, $\bar{y}$ is calculated from the following:

$$a = \sqrt{(x_n - \bar{x})^2 + (y_n - \bar{y})^2}$$

Eq. 33

and the component of that distance attributed to the atmospheric refraction is estimated by

$$\delta a \approx a\big(cot\alpha - cot(\beta + \alpha)\big)\Delta d$$

Eq. 34

where α is the angle of the laser beam from vertical (ellipsoid normal) and Δd is obtained using (Eq. 7). The value of β is obtained using

$$\beta = cos^{-1}\left(\frac{x_n(x_n - \bar{x}) + y_n(y_n - \bar{y})}{\sqrt{x_n^2 + y_n^2 + f^2}\sqrt{(x_n - \bar{x})^2 + (y_n - \bar{y})^2}}\right)$$

Eq. 35

The resulting image coordinate corrections are:

$$\Delta x_{atm} = -\frac{x_n - \bar{x}}{a}\delta a$$

Eq. 36

$$\Delta y_{atm} = -\frac{y_n - \bar{y}}{a}\delta a$$

Eq. 37

Combining the above atmospheric refraction corrections with the lens corrections (Eq. 24, Eq. 25) results in the following corrected values (*x'*,*y'*) for the image coordinates:

$$x' = \bar{x} + \Delta x_{lens} + \Delta x_{atm}$$

Eq. 38

$$y' = \bar{y} + \Delta y_{lens} + \Delta y_{atm}$$

Eq. 39

Taking into account all the image coordinate corrections needed for a frame-scanning system, if given pixel coordinates (r,c), corrected image coordinates would be calculated using Equations Eq. 14, Eq. 15, Eq. 16, Eq. 17, Eq. 19, Eq. 38 and Eq. 39.

4.2.3. Frame-scanner Sensor Equation

The frame-scanner equation takes on a similar form to ($r_{EP} = r_{ECEF} + M_{ECEF} M_{ELL} M_{VER} (M_{PLA} M_{GIM} 〖(M〗_{SEN} r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3) for the point-scanner sensor. However, the value of r_{SCA} (the range vector) must be adjusted to account for the frame geometry. The r_{SCA} value will be a function of the corrected image coordinates (*x', y'*), the focal length and the measured range from the ground point to the receiver focal plane.

Consider the example shown in **Error! Reference source not found.**. A LIDAR frame-scanning measurement of the ground point A produces an imaged point *a* at the receiver focal plane, with coordinates (*x', y'*). This results in a measured range represented by *R*, while *f* is the focal length and *L* is the location of the lens front nodal point. The value *r* is defined as follows:

$$r = \sqrt{(x')^2 + 〖(y〗')^2}$$

Eq. 40

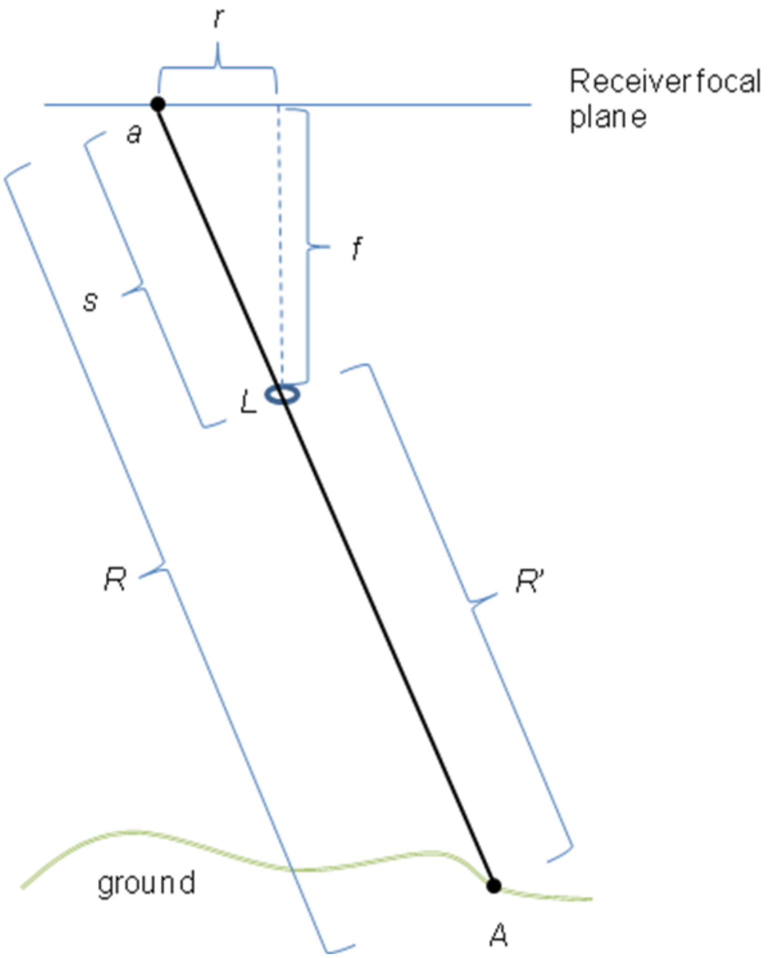


Figure 18. Frame receiver to ground geometry

It is necessary to transform the range measurement into the scanner coordinate system, which has its origin at the front nodal point of the lens and has its z-axis aligned with the lens optical axis (see **Error! Reference source not found.**). Two adjustments are necessary: subtracting *s* (the portion of the range measurement from the imaged point to the lens) from the range measurement *R* (resulting in *R'*); and correcting for the angular displacement of the range vector from the lens optical axis. The second correction is directly related to the image coordinates (*x', y'*), as shown by the equations below:

$$\boldsymbol{\theta} = \mathbf{arctan}\frac{\boldsymbol{x'}}{\boldsymbol{f}}\mathbf{n} \qquad \textbf{Eq. 41}$$

$$\boldsymbol{\varphi} = \mathbf{arctan}\frac{\boldsymbol{y'}}{\boldsymbol{f}} \qquad \textbf{Eq. 42}$$

where θ and φ are the x- and y-components of the angular displacement of the range vector from the lens optical axis. Corrections to the range measurement use the following equations:

$$s = \sqrt{f^2 + r^2}$$ **Eq. 43**

$$R' = R - s$$ **Eq. 44**

A rotation matrix *M* could then be constructed from θ and φ . The corrected value for r_{SCA} would then be

$$r_{SCA} = MR'$$ **Eq. 45**

$r_{EP} = r_{ECEF} + M_{ECEF} M_{ELL} M_{VER} (M_{PLA} M_{GIM} [\![(M]\!]_{SEN} r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3 could then be applied to calculate the geocentric ECEF coordinates of ground point *A*, imaged at image coordinates (*x’, y’*), using the value of r_{SCA} located above.

4.2.4. Collinearity Equations

The equations described in the previous section (4.2.3) are applied to obtain 3D ground coordinates of LIDAR points from a frame-scanner. However, depending on the application, it may be desirable to operate in image space (using *l, s* or *x’, y’*) rather than ground space (X, Y, Z). Therefore it becomes necessary to describe the relationship between image coordinates and ground coordinates, which is well described by the collinearity equations.

Deriving the relationship between image coordinates and the ground coordinates of the corresponding point on the Earth’s surface requires a common coordinate system, a process accomplished by translation and rotation from one coordinate system to the other. Extracting the object **A** from **Error! Reference source not found.**, the geometry is reduced to that shown in **Error! Reference source not found.**.

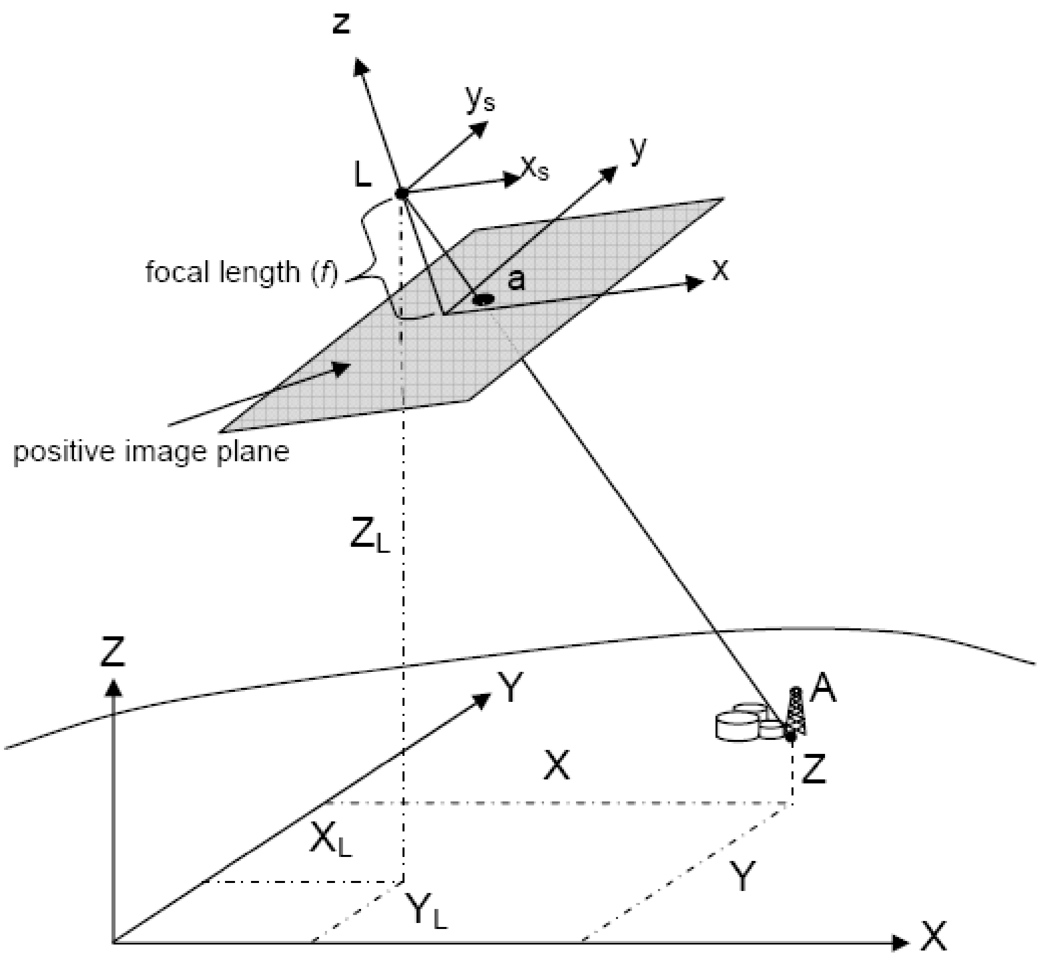


Figure 19. Collinearity of image point and corresponding ground point

Geometrically the sensor perspective center L, the "ideal" image point **a**, and the corresponding object point **A** are collinear. Note that the "ideal" image point is represented by image coordinates *after* having been corrected for all systematic effects (lens distortions, atmospheric refraction, etc.), as given in the preceding sections.

For two vectors to be collinear, one must be a scalar multiple of the other. Therefore, vectors from the perspective center L to the image point and object point, **a** and **A** respectively, are directly proportional. Further, in order to associate their components, these vector components must be defined with respect to the same coordinate system. Therefore, we define this association using the following equation:

$$\mathbf{a} = k\mathbf{M}\mathbf{A} \qquad \textbf{Eq. 46}$$

where k is a scalar multiplier and **M** is the orientation matrix that accounts for the rotations (roll, pitch, and yaw) required to place the Earth coordinate system parallel to the sensor coordinate system. Therefore, the collinearity conditions represented in the figure become:

$$\left[\begin{pmatrix}x\\y\\0\end{pmatrix}-\begin{pmatrix}0\\0\\f\end{pmatrix}\right]=kM\left[\begin{pmatrix}X\\Y\\Z\end{pmatrix}-\begin{pmatrix}X\\Y\\Z\end{pmatrix}_L\right]$$

Eq. 47

The orientation matrix **M** is the result of three sequence-dependent rotations:

$$M = M_K M_\varphi M_\omega = \begin{bmatrix}\cos K & \sin K & 0\\ -\sin K & \cos K & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\varphi & 0 & -\sin\varphi\\ 0 & 1 & 0\\ \sin\varphi & 0 & \cos\varphi\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\omega & \sin\omega\\ 0 & -\sin\omega & \cos\omega\end{bmatrix}$$

Eq. 48

where the rotation ω is about the X-axis (roll), φ is about the once rotated Y-axis (pitch), and *κ* is about the twice rotated Z-axis (yaw), the orientation matrix **M** becomes:

$$M = \begin{bmatrix}\cos\varphi\cos K & \cos\omega\sin K + \sin\omega\sin\varphi\cos K & \sin\omega\sin K - \cos\omega\sin\varphi\cos\\ -\cos\varphi\sin K & \cos\omega\cos K - \sin\omega\sin\varphi\sin K & \sin\omega\cos K + \cos\omega\sin\varphi\sin\\ \sin\varphi & -\sin\omega\cos\varphi & \cos\omega\cos\varphi\end{bmatrix}$$

Eq. 49

Using subscripts representing row and column for each entry in **M** results in the following representation:

$$M = \begin{bmatrix}M_{11} & M_{12} & M_{13}\\ M_{21} & M_{22} & M_{23}\\ M_{31} & M_{32} & M_{33}\end{bmatrix}$$

Eq. 50

Note that although the earlier derivation expressed coordinates with regard to the image plane ("negative" plane), the image point **a** in **Error! Reference source not found.** is represented by coordinates (x,y), whose relation is simply a mirror of the image plane. Thus the components of **a** will have opposite signs of their mirror components (x,y) as follows:

$$\bar{x} = -(x - x_0)$$

Eq. 51

$$\bar{y} = -(y - y_0)$$

Eq. 52

Eq. 47 represents three equations across the three rows of the matrices. Substituting Eq. 50 into Eq. 47 and dividing the first two equations by the third eliminates the *k* multiplier. Therefore, for any given object, its ECEF ground coordinates (X,Y,Z) are related to its image coordinates (x,y) by the following equations:

$$x = -f\left[\frac{M_{11}(X-X_L)+M_{12}(Y-Y_L)+M_{13}(Z-Z_L)}{M_{31}(X-X_L)+M_{32}(Y-Y_L)+M_{33}(Z-Z_L)}\right]$$

Eq. 53

$$y = -f\left[\frac{M_{21}(X-X_L)+M_{22}(Y-Y_L)+M_{23}(Z-Z_L)}{M_{31}(X-X_L)+M_{32}(Y-Y_L)+M_{33}(Z-Z_L)}\right]$$

Eq. 54

Note that (x,y) above represents the *corrected* pair, (*x',y'*), from Eq. 38 and Eq. 39. Also, the equations above rely upon the position and orientation of the sensor. The orientation is represented by the rotation matrix **M**, providing the rotation angles necessary to align the sensor coordinate system to the ECEF coordinate system (Section 3). Therefore **M** is simply the combination of rotation matrices provided in $r_{EP} = r_{ECEF} + M_{ECEF}M_{ELL}M_{VER}(M_{PLA}M_{GIM}\llbracket(M\rrbracket_{SEN}r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3, specifically

$$M = M_{ECEF}M_{ELL}M_{VER}M_{PLA} \qquad \textbf{Eq. 55}$$

Also, the position of the sensor r_L (X$_L$, Y$_L$, Z$_L$) can be obtained from ($r_{EP} = r_{ECEF} + M_{ECEF}M_{ELL}M_{VER}(M_{PLA}M_{GIM}\llbracket(M\rrbracket_{SEN}r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3) by setting the range vector r_{SCA} to zero, resulting in

$$r_L = r_{ECEF} + M_{ECEF}M_{ELL}M_{VER}(M_{PLA}\, r_{GIM} + r_{INS} + r_{GPS}) \qquad \textbf{Eq. 56}$$

5. Need for a LIDAR Error Model

Sections 3 and 4 described LIDAR sensor models and their associated equations. However, the question of why a sensor model is required and how it is used for LIDAR has not been addressed. The lidar product that most people are familiar with is a 3D point cloud of X,Y,Z points representing lidar returns. Thus, when the user receives the product, there is an inherent geopositioning aspect to it and it has already been rectified to ground space. However, sensor models are needed, even for preprocessed datasets. This section provides a general overview of the need for sensor models with LIDAR and is then followed with more technical sample use cases of sensor models for lidar.

What are the primary uses of sensor models for lidar? First, a sensor model must be used at the time of point cloud formulation when the raw measurements (the L0 data) are processed and combined to develop a 3D point cloud. These sensor models involve many of the ray tracing methods described earlier in this paper. While this paper provides a generalized overview of the concepts, the actual sensor model used during processing may be very sensor specific and all possible variations cannot be covered in this paper. Thus, this application of the sensor model, while potentially the most important, remains largely hidden to the end user.

To further complicate the issue, although the physical sensor model may be used to develop the initial point cloud, extensive post processing may be required subsequent to the initial cloud formulation to develop an improved product (less noise, improved accuracy, etc.) prior to the customer (analyst) receiving it.

A second use of sensor models for all sensor modalities is in data adjustment. It may be advantageous to adjust lidar data using parameters similar to the parameters used in the initial data formation. While these parameters may be similar to the physical parameters, they may be generalized parameters such as those proposed in the Universal Lidar Error Model (ULEM), thus developing a replacement model. This allows for rigorous data adjustments while still allowing many proprietary aspects of systems and data to remain hidden from the end user.

The final place that the sensor model may be used is in rigorous error propagation. For many applications, there is a need to estimate the accuracy of the final points on the ground in the ground reference frame. Precise geopositioning is the primary application that requires error propagation, but other applications also benefit from this ability to include data fusion.

The accurate development of predicted errors for ground points in a lidar dataset requires multiple inputs to include:

- The correct propagation of sensor errors into ground space
- Accounting for the uncertainty in data adjustment methods (either ground space or sensor space) into the final ground space coordinate.
- Accounting for any adjustment to a control or a reference source.
- Accounting for errors in the mensuration method and tool being employed.

Accurate error prediction requires consideration of all of the above and the influence each of the above has on the final coordinates will be dependent on the specific processing architecture. For example, there are linear mode lidar systems whose accuracy of the final product relies almost entirely on the accuracy of the originally collected sensor data along with considerations of initial calibration parameters. There are generally limited data adjustments performed on these datasets. However, there are other systems that rely very heavily on post processing adjustment techniques to achieve their final absolute accuracy. While there are adjustments being performed to improve absolute accuracy using sensor space parameters (the physical sensor model parameters discussed above), there are also many adjustments being performed in ground space. These adjustments are using local transformations and may have no direct tie to the physical sensor parameters.

To further complicate things, there are collections / processing performed that combine many individual lidar interrogations into a single output point in the final point cloud. One example of this is the process that may take place during the coincidence processing (denoising) of Geiger mode lidar data. While for many missions, the collection CONOP permits sensor space error propagation to be applied directly, for other cases tracing a point to a specific collection instance is difficult due to the potential of multiple collection instances for each output point.

Thus, while there are many benefits to using a rigorous sensor model for lidar, there are cases where the complexity of the point cloud collection and processing CONOPs preclude a straight forward definition of a sensor-space projective model. However, to be used for geopositioning, a method is still required that provides the user access to predicted errors for the dataset at the time of exploitation. Specifically, a method is needed to describe the spatial uncertainty relationships among X, Y, and Z at a ground point and the relationships between points necessary for relative error calculations. In these cases, the predicted errors may need to be developed upstream in the processing during the initial point cloud formulation and updated during processing to account for uncertainties in the adjustment process. This initial error calculation would still require the ability to propagate errors via a sensor model in the early processing stages. However, the data being carried downstream for exploitation would no longer include covariance information on sensor parameters; rather it would include a mechanism to describe the Pre-Computed ground space covariance matrix on a per point basis.

Thus, there are two primary sensor model exploitation methodologies currently being proposed for lidar. The first is an "On-Demand" model. This is the model that is consistent with current CSM methodologies where sensor metadata is carried all the way through downstream processing and used on an as-needed basis during exploitation. This has the advantage of providing the most rigor and flexibility during exploitation, but has the disadvantage of very complicated record keeping for certain systems / processing.

The second exploitation model is the "Pre-Computed" model. In this method, the sensor model is still used during upstream processing to develop a predicted error per point. However, it is the predicted errors in ground space and not the sensor covariance information that is carried downstream. Therefore during exploitation, the function calls retrieve ground space error covariance values versus calculating them on the fly.

The remainder of this paper focuses on both the “On-Demand” model and the “Precomputed” model.

6. Application of Sensor Model

This section provides an overview of the application of sensor models for lidar. Section **Error! Reference source not found.** uses the example of lidar registration to introduce the CSM functions related to lidar exploitation (Section **Error! Reference source not found.**) and then describes how these functions would be used in a block adjustment process (Section 6.2). For the “On-Demand” model these may be used during exploitation, while they would be used during the up-front processing for the “Pre-Computed” model. Section 6.3 provides an example of applying the CSM function calls for the common geopositioning practice of determining the ground covariance matrix at a point.

6.1 Key Components of Sensor Model

$r_{EP} = r_{ECEF} + M_{ECEF} M_{ELL} M_{VER} (M_{PLA} M_{GIM} [(M]_{SEN} r_{SCA} + r_{GIM}) + r_{INSMU} + r_{GPS})$ Eq. 3 and its ancillary equations from Section 4, depending on the system receiver geometry, can be applied to many aspects of LIDAR data for analysis. This section will discuss the access to components of the LIDAR sensor model and how the components can be used.

To exploit the sensor model, it is necessary to access various features of the model. However, particulars of a model can vary from sensor to sensor, and some of the mathematics may be proprietary. The Community Sensor Model (CSM) concept was developed to standardize access to sensor models. For a given class of sensors (e.g. frame imagery), key functions are used for relating sensed objects to sensor parameters. Sensor vendors then write and provide these functions so users can access the sensor model for a particular sensor without needing model information specific to the sensor. Any sensor within the same sensor class could then be accessed using the same key functions established for that class, assuming the key functions have been provided by the associated vendor.

In the case of a LIDAR sensor model, seven key functions will be described which help the user obtain the necessary information from the sensor model in order to perform tasks such as error propagation or parameter adjustment. Other CSM-based functions are available to access a LIDAR sensor model as well, but are not listed here since they are common across sensor classes. The key functions are:

1) ImageToGround()
2) GroundToImage()
3) ComputeSensorPartials()
4) ComputeGroundPartials()
5) ModelToGround()
6) GroundToModel()
7) GetPrecomputedCovariance()

Note that the list above reflects recommended changes and additions to CSM. This includes modifications to ComputeSensorPartials to expand the method domain to include ground space. It also includes the addition of two new methods called ModelToGround and GroundToModel. The “Model” coordinates are 3D ground-space coordinates calculated from a LIDAR sensor model but without any corrections applied from block adjustments, and the “ground” coordinates are adjusted model coordinates in the geocentric coordinate system. Finally, it includes the function GetPrecomputedCovariance that is used only for cases where a ground covariance matrix is precomputed and no sensor model is applied during final exploitation. All of these changes are described in more detail in the following section.

Instantiation of the key functions (excluding GetPrecomputedCovariance) is associated with a *state*. A state consists of the estimated sensor metadata values for a particular collection epoch. Therefore, when any of the functions are used, the state of the sensor has already been determined for that function call.

The key functions that are available for a given LIDAR dataset will depend on whether the data is represented in image space or ground space and whether it is using “On-Demand” or “Precomputed” methods. Image space is the native representation format for data collected from a frame scanner. Each frame consists of a raster of pixels (i.e. an image), with each pixel associated with a line/sample coordinate pair and having some type of height or range value. Ground space is the native representation format for data collected from a point scanner. A ground space dataset consists of 3D ground coordinates for each data point in some ground-referenced coordinate system.

Data represented in image space may be converted to ground space, since 3D coordinates can be calculated for each pixel in frame space. Therefore, a frame scanner may have its data available in image space or ground space. However, in practice, a point scanner only represents its data in ground space.

Following are descriptions of the key functions, followed by descriptions of how the functions can be used.

The table below provides an overview of the key functions, including the inputs and outputs for datasets provided in image space or ground space.

Table 1. Overview of Key Functions

Current (blue) functions, and proposed (green) changes and additions

Functions	Image Space	Ground Space
ImageToGround()	Input: line, sample Optional: image covariance Output: ground X, Y, Z Optional: ground covariance	N/A
GroundToImage()	Input: ground X, Y, Z Optional: ground covariance Output: line, sample Optional: image covariance	N/A
ComputeSensorPartials()	Input: ground X, Y, Z Optional: line, sample Output: $\partial_{line}/\partial_{parameter}$, $\partial_{sample}/\partial_{parameter}$	Input: ground X, Y, Z Output: partials of GroundToModel xyz wrt. sensor parameters. $\partial_{xyzXYZ}/\partial_{parameter}$
ComputeGroundPartials()	Input: ground X, Y, Z Output: ∂_l/∂_X , ∂_l/∂_Y , ∂_l/∂_Z , ∂_s/∂_X, ∂_s/∂_Y, ∂_s/∂_Z	Input: ground X, Y, Z Output: partials of GroundToModel xyz wrt. ground XYZ. $\partial_{xyz}/\partial_{XYZ}$
ModelToGround()	N/A	Input: model X, Y, Z Optional: multiple points, sensor parameter covariances Output: Adjusted X, Y, Z Optional: Ground covariance at multiple points
GroundToModel()	N/A	Input : ground X, Y, Z Optional : multiple points, ground covariances Output : model X, Y, Z Optional : model covariance at multiple points
GetPrecomputedCovariance	N/A	Input : X,Y,Z Optional : multiple points Output : Ground covariance at multiple points **

** Only applies for precomputed options. See description below.

Following are detailed descriptions of the key functions.

6.1.1. ImageToGround()

The ImageToGround() function returns the 3D ground coordinates in the associated XYZ geocentric (Earth Centered Earth Fixed) coordinate system for a given line and sample (*l, s*) of a LIDAR dataset in image space. This function is not applicable to data in ground space. If an optional image covariance matrix (2 by 2) is also provided as input, then the sensor model will use it and internally stored adjustable parameter covariance and range measurement variance to calculate an associated output ground covariance matrix (3 by 3) for the returned ground point coordinates.

6.1.2. GroundToImage()

The GroundToImage() function returns the line and sample (*l, s*) in image space for the given XYZ geocentric coordinates of a 3D ground point. This function is not applicable to data expressed only in ground space. If an optional ground covariance matrix (3 by 3) is provided as input, then the associated image covariance matrix (2 by 2) for the returned line/sample pair will be included in the output.

6.1.3. ComputeSensorPartials()

The ComputeSensorPartials() function returns partial derivatives of image line and sample (image space) or ground XYZ (ground space) with respect to a given sensor parameter. It can be executed in two different ways, depending on whether the partial derivatives are desired for data in image space or ground space. For both cases, the minimal input is XYZ coordinates of a 3D ground point and the index of a sensor parameter of interest. If image space partials are desired, an optional line/sample pair, associated with the ground XYZ coordinates, may also be provided as input (this allows for faster computation, since a call to GroundToImage() would be needed if the associated line/sample pair wasn't provided). For image space values, the output consists of partial derivatives of line and sample with respect to the input sensor parameter. For ground space values, the output consists of partial derivatives of model X, Y and Z with respect to the input sensor parameter. A typical set of adjustable parameters includes offsets in X, Y, and Z components of sensor position and offsets in roll, pitch, and yaw of platform or sensor attitude. Higher order terms in some of these components are typically added for time dependent systems. Furthermore, parameters such as range bias can be added to correct for calibration issues.

6.1.4. ComputeGroundPartials()

For image space, ComputeGroundPartials() returns partial derivatives of line and sample with respect to geocentric ground X, Y and Z values, resulting in a total of six partial derivatives. The input is the XYZ coordinates of a 3D ground point, and the output is the set of six partial derivatives. For ground space, ComputeGroundPartials() returns partial derivatives of model coordinates with respect to geocentric ground coordinates, resulting in a total of nine partial derivatives.

6.1.5. ModelToGround()

With the ModelToGround() function, given one or more model points as input, apply a transformation to the point(s) from the coordinate system of the original point cloud, e.g. UTM, to the ECEF coordinate system required by the CSM API. The transformation must also include the effects of the current values of all adjusted sensor parameters resulting from a block adjustment or from calibration values. Optional sensor parameter covariance can be included as input, which will provide ground covariance information as additional output. The sensor parameter covariance is associated with the set of adjustable parameters discussed in Section 6.1.3.

6.1.6. GroundToModel()

Given one or more geocentric ground points as input, determine the associated model coordinates of the points. This function is the inverse of the modelToGround() function discussed in Section 6.1.5. If an optional ground covariance matrix is provided as input, then the sensor model will employ full covariance propagation of it and the associated sensor parameter covariance matrix to output a full error covariance matrix of the returned 3D coordinates of the model points.

6.1.7. GetPrecomputedCovariance()

The GetPrecomputedCovariance() function is a proposed function exclusively for use with ground datasets using the "Pre-Computed" error methods. Given one or more model points as input, a full ground covariance matrix for the point or points of interest is returned. This is necessary because there is not a sensor state or a sensor covariance associated with this method during exploitation. Functionality to support ground-space adjustment may be added in the future. Note that this function is not used for datasets that are using an "On-Demand" error model. See ModelToGround() for more information on obtaining a ground covariance matrix when an "On-Demand" model is applied.

6.2 Application of On-Demand Sensor Model for Sensor Parameter Adjustment

One of the primary uses for a LIDAR sensor model is parameter adjustment. However, sensor parameter adjustment only applies to the On-Demand model since sensor covariance information is not carried in the Pre-Computed model. As an example using the On-Demand model, if provided multiple overlapping swaths of LIDAR data, one may want to adjust the exterior orientation parameters (X_L, Y_L, Z_L, ω, φ, *k*) for each swath, as well as the ground coordinates of common points, obtaining the best fit for the datasets in a least-squares sense. This is analogous to a bundle adjustment in photogrammetry (Mikhail, 2001).

A linearized least-squares estimation can be applied to perform the adjustment. The condition equations are shown in Eq. 57.

$$\underset{3\times 1}{F_{ij}} = groundToModel\left(\begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix}_j\right) - \begin{bmatrix} X_{\text{mod}} \\ Y_{\text{mod}} \\ Z_{\text{mod}} \end{bmatrix}_{ij} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Eq. 57

where *i* ranges from 1 to the number of swaths (*m*) and *j* ranges from 1 to the number of points (n). The term *groundToModel* is the CSM method described earlier. The first set of coordinates (with the *g* subscript) are the unknown adjusted ground coordinates of the tie points, while the second set (with the *mod* subscript) are the model, or swath, point coordinates of the tie-points.

The linearized form of the block adjustment, familiar from the photogrammetric applications, is given in Eq. 58.

$$\underset{3mn\times 1}{v} + \underset{3mn\times mu}{\dot{B}}\ \underset{mu\times 1}{\dot{\Delta}} + \underset{3mn\times 3n}{\ddot{B}}\ \underset{3n\times 1}{\ddot{\Delta}} = \underset{3mn\times 1}{f}$$

Eq. 58

where u is the number of sensor parameters. Derivation of the partial derivatives needed for the adjustment are shown in Eq. 59 and Eq. 60.

$$\underset{3\times u}{\dot{B}_{ij}} = \frac{\partial F_{ij}}{\partial S} = computeSensorPartials\left(\begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix}_j\right)$$

Eq. 59

$$\underset{3\times 3}{\ddot{B}_{ij}} = \frac{\partial F_{ij}}{\partial G_j} = computeGroundPartials\left(\begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix}_j\right)$$

Eq. 60

Eq. 59 is used for deriving the partial derivatives with respect to the sensor parameters (S), and uses the CSM method *computeSensorPartials* which is instantiated using the current values of the sensor parameters. Eq. 60 represents the partial derivatives with respect to the ground points.

The normal equations associated with the system of equations in Eq. 58 are given in Eq. 61. Inner constraints are necessary for the solution when ground control is not used. The two Δ terms are solved for, which contain corrections to the initial values for both the sensor parameters and the ground point coordinates.

$$\begin{bmatrix} \dot{B}^T\Sigma^{-1}\dot{B} + \Sigma_{SS}{}^{-1} & \dot{B}^T\Sigma^{-1}\ddot{B} \\ \ddot{B}^T\Sigma^{-1}\dot{B} & \ddot{B}^T\Sigma^{-1}\ddot{B} \end{bmatrix}^{-1} \begin{bmatrix} \dot{\Delta} \\ \ddot{\Delta} \end{bmatrix} = \begin{bmatrix} \dot{B}^T\Sigma^{-1}f - \Sigma_{SS}{}^{-1}f_S \\ \ddot{B}^T\Sigma^{-1}f \end{bmatrix}$$

Eq. 61

In Eq. 61 the following definitions apply: the *Σ_{SS}* matrix represents the *a priori* covariance matrix of the sensor parameters; the *f_S* vector represents the difference between current and observed values for the sensor parameters; the *Σ* matrix represents the tie-point measurement covariance values. If the tie points are measured manually, the repeatability in the measurement process could be used here. If the tie points are determined from an automated process, the resulting covariance from that process could populate this matrix. The input covariance matrix is shown in Eq. 62.

$$\underset{3mn\times 3mn}{\Sigma} = \begin{bmatrix} \underset{3\times 3}{\Sigma_1} & \underset{3\times 3}{0}\cdots & \underset{3\times 3}{0} \\ & \ddots & \vdots \\ sym & & \underset{3\times 3}{\Sigma_{mn}} \end{bmatrix}$$

Eq. 62

The steps used for solving the block adjustment are as follows:

1. Solve for the $\dot{\Delta}$ *and* $\ddot{\Delta}$ values in Eq. 61.
2. Update the sensor parameters and ground coordinates using the $\dot{\Delta}$ *and* $\ddot{\Delta}$ values.
3. Repeat steps 1 and 2 until convergence.
4. After convergence, use the inverse of the normal equations matrix (Eq. 61, 1st matrix) to obtain sensor parameter covariance and call *modelToGround* to obtain adjusted coordinate values and associated precision estimates for the LIDAR points.

6.3 Application of Sensor Model for Determination of Estimated Error

In all the metric applications of geospatial data, the quality of the extracted information is considered as important as the information itself. This is particularly true for geopositioning and targeting applications. Thus, it is critical that the user be provided methods to determine these quality values. This section begins with an overview of the desired error values and briefly discusses metadata required to obtain them. It then describes how to obtain these values for both the "On-Demand" and the "Pre-Computed" scenarios previously described.

The location of a target in the three-dimensional ground space is given either by its geodetic coordinates of longitude, λ, latitude, ϕ, and height (above the ellipsoid), h, or by a set of Cartesian coordinates X, Y, Z (currently more common for lidar). Although there are many ways to express the quality of the coordinates, the most fundamental is through the use of a *covariance matrix*. For example:

$$\sum_T = \begin{bmatrix} \sigma_X^2 & \sigma_{XY} & \sigma_{XZ} \\ \sigma_{XY} & \sigma_Y^2 & \sigma_{YZ} \\ \sigma_{XZ} & \sigma_{YZ} & \sigma_Z^2 \end{bmatrix}$$

Eq. 63

In which σ_X^2, σ_Y^2, σ_Z^2 are the marginal variances of the coordinates, and σ_{XY}, σ_{XZ}, σ_{YZ} are covariances between the coordinates, which reflect the correlation between them. The practice is often to reduce these six different numbers to only two, one expressing the quality of the horizontal position and the other the quality in the vertical position. The first is called *circular error*, or CE, and the second *linear error*, or LE. Both of these can be calculated at different probability levels, CE50 for 0.5 probability, CE90 for 0.9 probability, etc. Commonly used measures, particularly by NGA under "mapping standards," are CE90 and LE90. The CE90 value is derived from the 2 by 2 submatrix of **Σ** that relates to X *and Y*.

$$\begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix}$$

Eq. 64

The LE90 is calculated from σ_Z^2 . In these calculations, the correlation between the horizontal (*X,Y*) and vertical (*Z*) positions, as represented by σ_{XZ}, $\sigma_{YZ,}$ are ignored (i.e., assumed to be zero).

In order to have a realistic and reliable value for the estimated covariance matrix, Σ_T, of the geoposition, all the quantities that enter into calculating the coordinates *X,Y,Z* must have realistic and dependable variances and covariances. These latter values present the sensor modelers and exploiters with the most challenge. Sensor designers frequently do not provide any reasonable estimates of the expected errors associated with their sensor parameters. However, with well-calibrated sensors, it is usually reasonable to have the values of the needed sensor parameters as well as their quality in the form of standard deviations from calibration.

It may be that some of the sensor parameters are not reliably known. If such parameters are carried as adjustable parameters, then it is not critical to have good error estimates. These prior values can be approximate since, through the adjustment process, they would be refined through rigorous error propagation associated with least squares adjustment. These updated parameter covariances are, in turn, used in a rigorous propagation to produce the final covariance matrix, Σ_T, associated with each target.

The most difficulty is encountered when no adjustability is allowed and the information is based solely on the mission support data. In this case, if the input values for the quality of the parameters are either grossly in error, or non-existent, the propagated geolocation covariance matrix, Σ_T, can be considerably in error.

6.3.1. Application of On-Demand Sensor Model for Error Covariance

As discussed in Section 5, the On-Demand sensor model provides extended capabilities to support many operations, including data adjustment. However one of the basic functions, either pre or post data adjustment, is the ability to compute the error covariance matrix discussed in Section 6.3. With the On-Demand model, this is achieved by calling the ModeltoGround() function. In this case a model point, or a series of points, is passed into the function. The function then accesses the appropriate covariance information from the sensor and any data adjustment, and then propagates them into a ground covariance matrix. In addition to the ground points, the function then returns a 3N by 3N covariance matrix providing the errors for the N points passed into the function.

$$ModelToGround\left(\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix}_{i..j} \text{ and } \Sigma_{Sensor\ Parameters}\right) \Rightarrow \left(\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_{i..j} \text{ and } \begin{bmatrix} \underset{3\times3}{\Sigma_i} & & \underset{3\times3}{\Sigma_{ij}} \\ & \ddots & \\ sym & & \underset{3\times3}{\Sigma_j} \end{bmatrix}\right)$$

Eq. 65

Where model points and sensor parameters are passed in and ground points and a ground covariance are returned.

To make the function call and associated error propagation feasible, extensive metadata is required. These metadata requirements are illustrated in Section 7 below. Although this error propagation and associated adjustment may be unique per sensor, it is often possible to capture these parameters in a standardized replacement model that can then be used during exploitation. These standardized metadata items are discussed in section 7.2. Appendix A then provides an overview on how these standardized parameters would be exploited to compute an error covariance matrix during an application such as ModelToGround().

6.3.2. Application of Pre-Computed Sensor Model for Error Covariance

The application of the pre-computed sensor model is limited. It's primary function is to provide a ground covariance matrix at a point or a series of points so that the absolute and/or relative accuracies of the point(s) can be determined. This is useful for precise geopositioning and also for the exploitation of tie-points in fusion activities. This alone makes it very valuable for much of the community. However, its use in data adjustment and other advanced applications is limited.

Determining a predicted error using the pre-computed model is primarily the function call:

$$\boldsymbol{\Sigma} = GetPrecomputedCovariance\left(\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix}_{i..j}\right) = \begin{bmatrix} \underset{\mathbf{3\times3}}{\Sigma_\mathbf{i}} & \cdots & \underset{\mathbf{3\times3}}{\boldsymbol{\Sigma}_{ij}} \\ \vdots & \ddots & \vdots \\ sym & \cdots & \underset{\mathbf{3\times3}}{\boldsymbol{\Sigma}_\mathbf{j}} \end{bmatrix}$$

Eq. 66

In the above equation, model points are passed in and full error corvariance is returned.

Although the function calls to apply the pre-computed error model are very simple, the process required during the up-front processing to compute the error estimates can be very complicated and is specific to a particular LIDAR system and processing chain. Therefore, little time has been devoted in this paper to describing the methods used to compute these errors. However, there is work on-going within the NSG to define this process for several sensors and workflows.

Section 7.2.4 provides insight into one potential method to store this pre-computed error information within standard files. Appendix B provides an example of how this data would be used for multiple points during a GetPrecomputedCovariance call in order to compute a full ground covariance matrix for those points.

7. Sensor Metadata Requirements

The sections above described LIDAR systems, discussed the sensor equations required to generate 3-D points from a LIDAR system, and then discussed the design and application of CSM compliant sensor models and functions on LIDAR data. However, to make any of this work there is a need for appropriate metadata and this section describes these metadata requirements. It starts (7.1) by discussing metadata requirements for the initial creation of a LIDAR point cloud by looking at the metadata requirements to create the {X,Y,Z} coordinate for an individual LIDAR return. Although much of this may remain hidden to most users of lidar, the concepts are important for sensor parameter adjustment and error modeling. It then (7.2) looks at the metadata requirements for the application of a CSM compliant sensor model to a swath / block of data. The metadata required for “on-demand” error calculations are presented in section 7.2.3, followed by metadata required for “Pre-Computed” applications.

Note that there are currently multiple efforts ongoing to map LIDAR metadata, for example the LIDAR Conceptual Model and Metadata Dictionary (CMMD) for NGA InnoVision and the efforts of the Lidar Interoperability Focus Group (LiFG). Over time it will be necessary to consolidate and deconflict these various efforts. However, it is too soon to determine exactly what path will be taken by the NSG in the future and the basic geopositioning concepts will remain unchanged. So, this work is presented as a sample for reference, realizing that future applications of the methods may vary.

7.1 Metadata in Support of Sensor Equations

A compilation of the whiskbroom model parameters (associated with linear mode flying spot scanners) and array model parameters are given in the following tables. Table 2 provides the fundamental data set that the sensor must provide such that the sensor models described in section 4 can be applied to single points, and, therefore, those parameters specifically required to establish the final point cloud. Distinction between what the sensor provides and the entire collection system (including the platform and other external sources of data) is important, because processing / exploitation tools must be designed to retrieve the appropriate data and from the appropriate source. For example, a sensor may not be expected to provide its orientation with respect to WGS-84, but rather to the platform from which it operates which presumably would produce data with respect to WGS-84.

Table **3** lists those parameters required of the platform to support orientation of the sensor such that conversion between image and object coordinates is possible.

Table 2. Sensor model type definition and parameters

(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	Parameter	***Definition***	**Units**	***Ob FS***	***Ob PS***	***Description***
1	Sensor Type	Classification indicative of the characteristics of the collection device.	N/A	M 0-2	M 0-2	STANAG 7023 further defines types (e.g., $01 FRAMING, $02 LINESCAN, $05 STEP FRAME", etc.). NOTE: LIDAR is currently not included in STANAG 7023. If possible a better Sensor Type would include: LIDAR FRAMING, LIDAR LINESCAN, and LIDAR STEP FRAME.
2	Number of Columns in Sensor Array	The number of columns in the sensor array (N_y). (unitless)	integer	M 0-2	X	The number of columns in the array. For LIDAR, this is the number of possible ranging elements in the column direction per pulse. Excluded for linear mode/whiskbroom.
3	Sensor Array Width	Aggregate dimension of the sensor array in the y-direction	Millimeters or radians	C 0-2	X	Conditional because it may not be required for linear whiskbroom sensors and, when needed, this could also be calculated from array size and spacing.
4	Column Spacing d_y	Column spacing, d_y, measured at the center of the image; distance in the image plane between adjacent pixels within a row.	Millimeters	M 0-2	X	NITF definition, STDI-0002, ACFTB, "COL_SPACING", includes angular and linear measurement methods.
5	Number of Rows in Sensor Array	The number of rows in the sensor array. (unitless)	integer	M 0-2	X	The number of rows in the array. For LIDAR, this is the number of possible ranging elements the row direction per pulse. Excluded for linear model/whiskbroom.
6	Row Spacing	Row spacing, d_x, measured at the center of the image; distance in the image plane between corresponding pixels of adjacent columns.	Millimeters	M 0-2	X	NITF, STDI-0002 ACFTB, "ROW_SPACING", includes angular and linear measurement methods.
7	Collection Start Time	The date and time at the start of the LIDAR pulse.	TRE code	M	M	The time of the LIDAR pulse emission
8	Collection	The date and time that	TRE	M	M	The time of the LIDAR pulse is returned

	Stop Time	the emitted pulse is received on a sensor detector for a given pulse	code			
9	Sensor Position, X Vector Component	X component of the offset vector; x-axis measurement, mm, of the vector offset from the origin of the sensor mounting frame, e.g. gimbal platform to the origin of the sensor perspective center, L.	Millimeters	M 0,1	M 0,1	Offset vector described the position of the sensor perspective center relative to a gimbal position (if any), which, in turn may be referenced to the platform coordinate system; or the offset may be given directly to the platform coordinate system, if known.
10	Sensor Position, Y Vector Component	Y component of the offset vector, y-axis measurement, mm, of the vector offset from the origin of the sensor mounting frame, e.g. gimbal platform to the origin of the sensor perspective center, L.	Millimeters	M 0,1	M 0,1	See Sensor Position, X Vector Component
11	Sensor Position, Z Vector Component	Z component of the offset vector, z-axis measurement, mm, of the vector offset from the origin of the sensor mounting frame, e.g. gimbal platform to the origin of the sensor perspective center, L.	Millimeters	M 0,1	M 0,1	See Sensor Position, X Vector Component
12	Sensor Rotation about Z-axis	Rotation of the sensor at pulse time (t) in the xy plane of the sensor reference frame; positive when positive +x axis rotates directly towards +y axis. (radians)	radians	M 0	M 0	Derived value computed at a given time by Kalman filtering of sensor image acquisition time (t) with platform attitude time (t). Reference may be made to either the gimbal mounting or to platform reference system; but must be specified. If these rotation angles are gimbal mounting angles, classic, this development transforms them into the required sequential Euler angles.
13	Sensor Rotation about Y-	Rotation of the sensor at pulse time (t) in the xz plane of the sensor	radians	M 0	M 0	See Sensor Rotation about Z-axis.

	axis	reference frame; positive when positive +z axis rotates directly towards +x axis. (radians)				
14	Sensor Rotation about X-axis	Rotation of the sensor at pulse time (t) in the yz plane of the sensor reference frame; positive when positive +y axis rotates directly towards +z axis. (radians)	radians	M 0	M 0	See Sensor Rotation about Z-axis.
15	Sensor Focal Length	*f*, lens focal length; effective distance from optical lens to sensor element(s).	millimeters	M 0-2	C 0	Similar to STDI-0002 TRE ACFTB, Focal_length, page 79, Table 8-6; “effective distance from optical lens to sensor element(s), used when either ROW_SPACING_UNITS or COL_SPACING_UNITS indicates μ-radians. 999.99 indicates focal length is not available or not applicable to this sensor. NOTE: Depending on the model, focal length values may or may not be used for linear / whiskbroom scanners.
16	Sensor Focal Length Flag	Value that defines if the provided focal length is a calibrated focal length.	logical	M 0-2	C 0	Y(es) / N(o) or 1 / 0 value that indicates Calibrated or Not-calibrated focal length value is provided.
17	Sensor Focal Length Adjustment	Refinement (Δf) resulting from self-calibration operation	millimeters	C 0-2	C 0	Nominally a single value for a data set collection, however refinement may be defined for each segment of the total image collection. Conditional on the implementation of a self-calibration operation in the software.
18	Principal point off-set, x-axis	x-coordinate with respect to the sensor coordinate system, of the foot of the perpendicular dropped from perspective center (focal point) of the sensor lens onto the collection array. (frame sensor)	millimeters	M 0-2	X	Nominally a single value for a data set collection. Initially this approximation is based on sensor component quality that is refined in the self-calibration / geopositioning operation. As a coordinate, this term includes magnitude and direction (i.e., positive/negative x). Conditional when this is replaced with calibration, measured, or look up table data. NITF and STDI do not specifically address point off-sets. NOTE: This term is not used in the linear mode / whiskbroom solution.
19	Principal point off-set, y-axis	y-coordinate with respect to the sensor coordinate system, of the foot of the perpendicular dropped from perspective center	millimeters	M 0-2	X	Nominally a single value for a data set collection. Initially this approximation is based on sensor component quality that is refined in the self-calibration / geopositioning operation. As a coordinate, this term includes magnitude and direction (i.e., positive/negative y). Conditional when this is replaced with calibration, measured, or

		(focal point) of the sensor lens onto the center of the collection array. (frame, pushbroom, whiskbroom)				look up table data. NITF and STDI do not specifically address point off-sets. NOTE: This term is not used in the linear mode / whiskbroom solution.
20	Principal Point offset covariance data	Covariance data of principal point offsets (mm^2).	millimeters squared	O 0-2	X	In practice, of such small magnitude so as can be ignored.
21	Sensor position and attitude accuracy variance and covariance data	$\begin{array}{cccccc} \sigma^2_{X_L} & \sigma_{X_LY_L} & \sigma_{X_LZ_L} & \sigma_{X_L\omega} & \sigma_{X_L\varphi} & \sigma_{X_L\kappa} \\ & \sigma^2_{Y_L} & \sigma_{Y_LZ_L} & \sigma_{Y_L\omega} & \sigma_{Y_L\varphi} & \sigma_{Y_L\kappa} \\ & & \sigma^2_{Z_L} & \sigma_{Z_L\omega} & \sigma_{Z_L\varphi} & \sigma_{Z_L\kappa} \\ & & & \sigma^2_{\omega_L} & \sigma_{\omega\varphi} & \sigma_{\omega\kappa} \\ & & & & \sigma^2_{\varphi_L} & \sigma_{\varphi\kappa} \\ & & & & & \sigma^2_{\kappa_L} \end{array}$ Symmetric matrix Variance (sigma^2) and covariance data for position (X_L,Y_L,Z_L), and attitude (roll, pitch, yaw).	Millimeters squared / radians squared	M 0,1 C 2,3	M 0,1 C 2,3	Initially these are values provided from the GPS and IMU components, but may be refined in data adjustment operations. Conditional only if timestamps can be carried forward in L2 and L3 data.
22	Focal length accuracy variance data	σ_f^2 Variance (mm^2) data for focal length.	Millimeters squared	M 0-2	C 0-2	Single value from sensor calibration or data adjustment operation. May not apply to linear mode / whiskbroom scanners. For whiskbroom / linear scanners, the need for this value is conditional only if the focal length is used in the point determination.
23	Lens radial distortion coefficients	k_1 (mm^{-2}), k_2 (mm^{-4}), k_3 (mm^{-6}), lens radial distortion coefficients	various reciprocal units	C 0-2	X	Single set of values either from sensor calibration or geopositioning operation Alternatively, may be replaced with calibration, measured, or look up table data. NITF and STDI-0002 do not specifically address distortion factors. k_3 may be ignored, in most situations.
24	Lens radial distortion (k1,k2,k3) covariance data	Covariance data of lens radial distortion.		O	X	In practice, of such small magnitude, so as can be ignored.
25	Decentering lens correction	p_1(mm^{-2}), p_2(mm^{-2})	various reciprocal	O	X	Single set of values either from sensor calibration or geopositioning operation Alternatively, may be replaced with calibration, measured, or look up table data.

	coefficients		units			NITF and STDI-0002 do not specifically address distortion factors.
26	Decentering lens correction (p1,p2) covariance data	Covariance data of decentering lens correction coefficients.		O	X	In practice, of such small magnitude, so as can be ignored.
27	Atmospheric correction (Δd) by data layer	Correction to account for bending of the image ray path as a result of atmospheric effects	Micro-radians	C 0-2	C 0-2	Adjustment to compensate for the bending in the image ray path from object to image due to atmospheric effects. Multiple data layers can be defined so the parameter has an index of I= 1, …n
28	Atmospheric correction data layer top height	Upper boundary altitude value for data layer I	meters	C 0-2	C 0-2	Sets the upper bound for the specific atmospheric correction value for data layer I
29	Atmospheric correction data layer bottom height	Lower boundary altitude value for data layer I	Meters	C 0-2	C 0-2	Sets the lower bound for the specific atmospheric corrections value for data layer I
30	Atmospheric correction algorithm name	Name of algorithm used to compute data layer I correction	String	C 0-2	C 0-2	Defines the specific algorithm used in the computation Conditional on the use of a correction
31	Atmospheric correction algorithm version	Version label for the algorithm used to compute data layer I correction	String	C 0-2	C 0-2	Defines the specific version of the algorithm use in the computation
32	Swath Field of View (FOV)	Nominal object total field of view of the sensor using the complete range of angles from which the incident radiation can be collected by the detector array	degree	M 0-2	M 0-2	The field of view being used for a given collection, defined in degrees.
33	Instantaneous Field of View	The object field of view of the detector array in the focal plane at time (t)	degrees	M 0-2	X	Normally measured in degrees.
34	Scan	Actual scan angle values	Milli-	M 0	M 0	Normally measured in milli-radians. May be multiple angle values

	Angles at pulse time (t)	at sensor time t for pixel array	radians			depending on the system.
35	Time (t)	Time value	microseconds	M 0,1 C 2,3	M 0,1 C 2,3	Value used to interpolate platform and sensor location and attitude. Value also used to determine whiskbroom scan angle. Conditional if time can be carried forward in L2 and L3 data.
36	GPS Lever arm offset	Vector from GPS to IMU described in either x, y, z components or by magnitude and two rotations.	millimeters	C 0-2	C 0-2	Conditional on platform geolocation at scan line time being sent; if platform geolocation provided wrt IMU, this lever arm unnecessary.
37	IMU Lever arm offset	Vector from IMU to the sensor reference point described as x,y,z components or by magnitude and two rotations	millimeters	C 0-2	C 0-2	Conditional on platform geolocation at scan line time being sent; if platform geolocation provided wrt sensor reference point, this lever arm is unnecessary.

Table 3. Collection platform parameters

(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Comments*
1	Platform Location Time $P_{(t)}$	UTC time when platform location data is acquired.	Micro-seconds	M 0,1 C 2	M 0,1 C 2	Provides data to correlate platform location to sensor acquisition. Conditional on Collection Start Time being simultaneously collected with image data, to provide necessary orientation of sensor/platform/Earth reference.
2	Platform geolocation at return time $P_{(t)}$	The position of the platform at return time (t) in XYZ ECEF coordinates.	meters	M 0,1 C 2,3	M 0,1 C 2,3	Conditional on sensor position (longitude, latitude) being sent, only if sensor position is relative to an absolute reference. Center of navigation defined wrt the local NED platform coordinate frame, then related to an ECEF reference. Consideration should be given to allowing the reference system to be defined when the location values are provided. This would be consistent with the Transducer Markup Language OpenGIS® Implementation Specification (OGC® 06-010r6), which requires the source, values, and all associated information to be provided to uniquely define location data, instead of mandating a specific reference system. Conditional obligation if timestamp can be carried forward in L2 and L3 data.
3	Platform Attitude Determination Time at return time $P_{(t)}$	UTC time when platform attitude (IMU) data is acquired.	Micro-seconds	M 0	M 0	Provides data to associate platform attitude to sensor attitude at range acquisition. Conditional on attitude data simultaneously collected with range timing data, or platform location time provided.
4	Platform true heading at return time $P_{(t)}$	Platform heading relative to true north. (positive from north to east)	radians	M 0	M 0	Conditional on sensor position and rotation data available directly when given within an absolute reference. Added to STANAG definition, "(positive from north to east)". Alternatively, true heading not required if platform yaw is given.
5	Platform pitch at return time $P_{(t)}$	Rotation about platform local y-axis (Y_a), positive nose-up; 0.0 equals platform z-axis (Z_a) aligned to Nadir, limited	radians	M 0	M 0	Conditional on sensor position and rotation data available directly when given within an absolute reference. Consistent with STANAG 7023, paragraph A-6.1; added "limited" values to definition. Alternatively, true heading not required if platform pitch (Item 3) is given.

		to values between +/- 90degrees.				
6	Platform roll at return time $P_{(t)}$	Rotation about platform local x-axis (X_a). Positive port right wing up. (degrees)	radians	M 0	M 0	Conditional on sensor position and rotation data available directly when given within an absolute reference. Consistent with STANAG 7023, paragraph A-6.1. Alternatively, true heading not required if platform roll (Item 3) is given.
7	Platform true airspeed	Platform true airspeed at data acquisition time (t) (m/second)	Meters/ second	O	O	Optional value that may not be used by the sensor model.
8	Platform ground speed	Platform velocity over the ground at data acquisition time (t) (m/second)	Meters/ second	O	O	Optional value that may not be used by the sensor model.

7.2 Metadata in Support of CSM Operations

The section above described the types of metadata required to take the raw sensor observations and covert them into a single 3-D point. However, this process and the metadata required to perform this process can be very sensor and processor specific and often involves proprietary information / data. At the present time, this processing will often be performed by the data provider and the LIDAR data will then be provided to the user as a point cloud. This could be a point cloud consisting of a series of individual swaths or it could be a series of swaths combined together. Regardless of the format, there is still data analysis and adjustments that the user may wish to perform on this data and this data adjustment may be performed using a LIDAR specific CSM model. This section describes the data / metadata that would be required to employ a CSM model for the functions described in section 5. This section also addresses the needed metadata for storing pre-computed error, as described in section 6.

During the production of the following metadata names and definitions, an attempt was made to harmonize these values with those defined in the ASPRS LAS 1.2 Format Specification (ASPRS, 2008).

7.2.1. Dataset-Common Information

The values below describe dataset-common information that must be stored per dataset in order to properly access the point cloud (in ground space coordinates) and apply the CSM functions described in section 5.

Table 4. Dataset-Common Information
(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Description*
1	Sensor Type	Classification indicative of the characteristics of the collection device.	N/A	M 1-3	M 1-3	STANAG 7023 further defines types (e.g., $01 FRAMING, $02 LINESCAN, $05 STEP FRAME", etc.). NOTE: LIDAR is currently not included in STANAG 7023. If possible a better Sensor Type would include: LIDAR FRAMING, LIDAR LINESCAN, and LIDAR STEP FRAME
2	Collection Start Time	The date and time at the start of the LIDAR collection.	TRE code	M 1-3	M 1-3	The time of the start of the LIDAR data associated with this file
3	X scale Factor	Scale factor for X coordinate of the point.	Unitless	M 1-3	M 1-3	Value used to scale the X record value stored per point prior to applying the X offset to determine the X coordinate of a point.
4	Y Scale Factor	Scale factor for Y coordinate of the point.	Unitless	M 1-3	M 1-3	Value used to scale the Y record long value stored per point prior to applying the Y offset to determine the Y coordinate of a point.
5	Z Scale Factor	Scale factor for Z coordinate of the point.	Unitless	M 1-3	M 1-3	Value used to scale the Z record long value stored per point prior to applying the Z offset to determine the Z coordinate of a point.
6	X Offset	Point record offset in the X direction	Units of point cloud	M 1-3	M 1-3	An offset applied to the product of the X record and X scale factor to obtain the X coordinate for a point
7	Y Offset	Point record offset in the Y direction	Units of point cloud	M 1-3	M 1-3	An offset applied to the product of the Y record and Y scale factor to obtain the Y coordinate for a point
8	Z Offset	Point record offset in the Z direction	Units of point cloud	M 1-3	M 1-3	An offset applied to the product of the Z record and Z scale factor to obtain the Z coordinate for a point

7.2.2. Point Record Information

The values below describe the point record information that must be stored on a per point basis in order to apply the CSM functions described in section 5.

Table 5. Point Record Information
(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Description*
1	Point X record	The X position of a specific point in the specified coordinate system.	Units of point cloud	M 1-3	M 1-3	The X position of the LIDAR returns stored per point record. This value is used in combination with the X Scale Factor and X Offset to determine the X coordinate of a given point.
2	Point Y record	The Y position of a specific point in the specified coordinate system.	Units of point cloud	M 1-3	M 1-3	The Y position of the LIDAR returns stored per point record. This value is used in combination with the Y Scale Factor and Y Offset to determine the Y coordinate of a given point.
3	Point Z record	The Z position of a specific point in the specified coordinate system.	Units of point cloud	M 1-3	M 1-3	The Z position of the LIDAR returns stored per point record. This value is used in combination with the Z Scale Factor and X Offset to determine the Z coordinate of a given point.
4	Intensity	The intensity of the return recorded by the system	Unitless	C 1-3	C 1-3	An integer value that represents the intensity of the energy returning to the system on a given return for a specified pulse. Conditional if available in system.
5	Range Uncertainty	Uncertainty in the range dimension	meters	C 1-2	C 1-2	The uncertainty in the range dimension for a specific LIDAR return. Although necessary, this is conditional in this table because it could be stored and/or calculated using several methods. It could be pre-calculated per point (as shown here), it could be considered constant, it could be stored as a function of time, or it could be calculated on the fly as needed.
6	Time (t)	Time value	microseconds	M 1-2	M 1-2	Time associated with the specific return of the sensor. Will be used to determine other sensor parameters at a given time.
7	Covariance Set ID	A unique numerical identifier for a covariance matrix pertaining to the ground error for a set of points	Unitless	C 1-3	C 1-3	Conditional on the implementation of pre-computed error modeling.

Note that the data in Tables 4 and 5 will be combined to generate ground space coordinates for points in the 3D point cloud. The ground space coordinates are derived as follows:

$X_{\text{coordinate}} = (\text{Point } X_{\text{record}} * X_{\text{scale Factor}}) + X_{\text{offset}}$ $Y_{\text{coordinate}} = (\text{Point } Y_{\text{record}} * Y_{\text{scale Factor}}) + Y_{\text{offset}}$ $Z_{\text{coordinate}} = (\text{Point } Z_{\text{record}} * Z_{\text{scale Factor}}) + Z_{\text{offset}}$

7.2.3. Modeled Uncertainty Information

The values below describe sensor / collection information that must be available in order to calculate the uncertainty at a given point using the functions described in section 5. It may not be necessary to store these values on a per point record basis. This would depend on the sensor being used and the time scale over which the values discussed below change.

7.2.3.1. Platform Trajectory

Although it does not have to be the exact trajectory, there is a need to know the approximate trajectory of the sensor so that the approximate location of the platform for a given point can be calculated, which in turn allows the calculation of approximate sensor LOS angles. The sample rate of this trajectory may vary based on platform speed and platform motion. At a minimum, the following values are needed:

Table 6. Platform Trajectory Information
(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Comments*
1	Sensor Location Time t_p	UTC time for a specific platform location	Micro-seconds	M 1 C 2,3	M 1 C 2,3	Provides data to correlate sensor location to sensor acquisition. Conditional if timestamps available in L2 and L3.
2	Sensor Geolocation at time t	The position of the sensor at time (t) in XYZ ECEF coordinates	Meters or Feet	M 1 C 2,3	M 1 C 2,3	The sensor position relative to an absolute reference at a given time t. The platform position and orientation angles have been combined with the sensor pointing information to obtain the position of the sensor reference point. Conditional if timestamps available in L2 and L3.

4	Sensor position accuracy and covariance data at time t	$\begin{matrix} \sigma^2_{X_L} & \sigma_{X_LY_L} & \sigma_{X_LZ_L} \\ & \sigma^2_{Y_L} & \sigma_{Y_LZ_L} \\ & & \sigma^2_{Z_L} \end{matrix}$ Symmetric matrix Variance (sigma^2) and covariance data for sensor position (X_L,Y_L,Z_L),	meters squared	M 1 C 2,3	M 1 C 2,3	Initially these are values provided from the GPS, IMU, and pointing components, but may be refined in data adjustment operations. Conditional if timestamps available in L2 and L3.

7.2.3.2. Sensor Line of Sight (LOS) Uncertainty

In addition to the trajectory, there is a need to know the line of sight (LOS) uncertainty of the sensor as a function of time. The sample rate of this line of sight function may vary based on platform speed and platform motion. Please note that this is not meant to store the individual components that contribute to LOS uncertainty as these may be complicated and proprietary. Rather, this provides the data provider and the data exploiter a method to determine the combined LOS uncertainty at a specified reference time. At a minimum, the following values are needed:

Table 7. Sensor LOS Uncertainty Information

(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Comments*
1	Sensor Line of Sight Time $P_{(t)}$	UTC time for a specific line of sight uncertainty information	Micro-seconds	M 1 C 2,3	M 1 C 2,3	Provides data to correlate sensor line of sight to sensor acquisition. Conditional if timestamps available in L2 and L3.
2	Sensor line of sight accuracy variance and covariance data at time t.	$\begin{bmatrix} \sigma_{\omega}^{2} & \sigma_{\varpi\varphi} & \sigma_{\varpi\kappa} \\ & \sigma_{\varphi}^{2} & \sigma_{\varphi\kappa} \\ sym. & & \sigma_{\kappa}^{2} \end{bmatrix}$ Symmetric matrix Variance (sigma^2) and covariance data for line of sight.	radians	M 1 C 2,3	M 1 C 2,3	Initially these are values calculated from the combination of IMU and pointing components, but may be refined in data adjustment operations. Conditional if timestamps available in L2 and L3.

7.2.3.3. Parameter Decorrelation Values

When calculating relative errors between points, in addition to the trajectory and LOS uncertainties, there is a need to know how the uncertainty values are correlated as a function of time. Points collected close together in time would be expected to be highly correlated and this correlation would decrease as the time separation increases.

Table 8. Parameter Decorrelation Values Information
(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Comments*
1	Sensor Position Decorrelation Parameter	A parameter (β) used in the decorrelation function ($\rho = e^{-\beta\|t_2 - t_1\|}$) as it applies to sensor horizontal position.	Unitless	O	O	Used to determine how the sensor position becomes decorrelated over time. Marked as optional, but may be necessary for accurate representation of relative accuracies over short distances.
2	Sensor LOS Decorrelation Parameter	A parameter (β) used in the decorrelation function ($\rho = e^{-\beta\|t_2 - t_1\|}$) as it applies to sensor line of sight	Unitless	O	O	Used to determine how the sensor line of sight becomes decorrelated over time. Marked as optional, but may be necessary for accurate representation
3	Sensor range Decorrelation Parameter	A parameter (β) used in the decorrelation function ($\rho = e^{-\beta\|t_2 - t_1\|}$) as it applies to sensor range	Unitless	O	O	Used to determine how the sensor range becomes decorrelated over time. Marked as optional, but may be necessary for accurate representation

7.2.4. Pre-computed Uncertainty Information

In the case where ground point uncertainties would be pre-computed, the following metadata would be necessary. This information would not need to be stored per point, but would apply to a set of points with similar uncertainty character.

7.2.4.1. Ground Covariance Information

Pre-computed ground uncertainties, for a given set of points, would be represented by a 3 by 3 covariance matrix, as described in the following table:

Table 9. Pre-computed Ground Covariance Information
(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Comments*
1	Covariance Set ID	Unique numerical identifier for the ground covariance uncertainty matrix associated with a particular set of points.	Unitless	C 1,2,3	C 1,2,3	Conditional on implementation of pre-computed error modeling.
2	Ground covariance uncertainties.	$\begin{matrix} \sigma^2_X & \sigma_{XY} & \sigma_{XZ} \\ & \sigma^2_Y & \sigma_{YZ} \\ & & \sigma^2_Z \end{matrix}$ Symmetric matrix Variance (sigma^2) and covariance data for ground uncertainty in ECEF coordinate system.	meters squared	C 1,2,3	C 1,2,3	Conditional on implementation of pre-computed error modeling.

7.2.4.2. Ground Uncertainty Decorrelation

It is expected that errors would be correlated for adjacently collected points, and the correlation would decrease with distance. The matrix in the following table addresses the decorrelation values that would be needed to calculate the error correlation between two points.

Table 10. Ground Uncertainty Decorrelation
(Obligation: M - Mandatory, C - Conditional, O – Optional, X – excluded or not needed, TBR – To be resolved, Ob PS – Obligation Point Scanning System, Ob FS – Obligation Frame Scanning System)

ID	*Parameter*	*Definition*	*Units*	*Ob PS*	*Ob FS*	*Comments*
1	Ground uncertainty decorrelation values	β_U β_V β_W Decorrelation exponential coefficient values used to calculate the error correlation between two ground points.	1/meters	C 1,2,3	C 1,2,3	Conditional on implementation of pre-computed error modeling.
2		ω, φ, κ Sequential rotation angles about the X, Y, Z axes to rotate to the U,V,W coordinate system for decorrelation.	radians	C 1,2,3	C 1,2,3	Conditional on implementation of pre-computed error modeling. These rotation angles are used to account for situations in which the data becomes decorrelated by distance, but when the decorrelation does not occur in cardinal ground space directions (X,Y,Z). These angles allow the decorrelation to refer to an alternate coordinate system (U, V, W).

References

1. Aull, Brian F et al., Geiger-mode Avalanche Photodiodes for Three-Dimensional Imaging", MIT Lincoln Laboratory Journal, Vol. 13, No. 2, 2002, pp. 335-350.

2. Anderson, James M and Mikhail, Edward M. Surveying Theory and Practice. McGraw-Hill Companies, Inc, 1998.

3. ASPRS LAS 1.2 Format Specification, September 2, 2008.

4. ASTM E2544-07a, "Standard Terminology for Three-Dimensional (3-D) Imaging Systems", ASTM International

5. Baltsavias, E.P. Airborne Laser Scanning: Basic Relations and Formulas. ISPRS Journal of Photogrammetry & Remote Sensing 54_1999.199–214

6. Brenner, Claus. Aerial Laser Scanning. International Summer School "Digital Recording and 3D modeling". April 2006

7. Chauve, A. "Processing Full-Waveform LIDAR Data: Modeling Raw Signals", ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo, September 12-14, 2007, Finland.

8. Community Sensor Model (CSM) Technical Requirements Document, Version 3.0, December 15, 2005.

9. DMA-TR-8400. DMA Technical Report: Error Theory as Applied to Mapping, Charting, and Geodesy

10. Federal Geographic Data Committee (FGDC) Document Number FGDC-STD-012-2002, Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata.

11. Goshtasby, A, 2-D and 3-D Image Registration For Medical, Remote Sensing, And Industrial Applications, 2005. John Wiley & Sons, Inc.

12. Gyer, M.S., "Methods for Computing Photogrammetric Refraction Corrections for Vertical and Oblique Photographs," Photogrammetric Engineering and Remote Sensing, Vol. 62, No. 3, March 1996, 301-310.

13. ISO/IEC 12087-5, Information Technology -- Computer graphics and image processing -- Image Processing and Interchange (IPI) -- Functional specification -- Part 5: Basic Image Interchange Format (BIIF), 1998.

14. ISO/IEC 2382-1, Information Technology -- Vocabulary -- Part 1: Fundamental terms, 1993.

15. ISO/IEC 2382-17, Information Technology -- Vocabulary -- Part 17: Databases, 1999.

16. ISO TC/211 211n1197, 19101 Geographic information – Reference model, as sent to the ISO Central Secretariat, for registration as FDIS, December 3, 2001.

17. ISO TC/211 211n2047, Text for ISO 19111 Geographic Information - Spatial referencing by coordinates, as sent to the ISO Central Secretariat for issuing as FDIS, July 17, 2006.

18. ISO TC/211 211n2171, Text for final CD 19115-2, Geographic information - Metadata - Part 2: Extensions for imagery and gridded data, March 8, 2007.

19. ISO TC211 211n1017, Draft review summary from stage 0 of project 19124, Geographic information - Imagery and gridded data components, December 1, 2000.

20. ISO TC211 211n1869, New Work Item proposal and PDTS 19129 Geographic information - Imagery, gridded and coverage data framework, July 14, 2005.

21. ISO/TS 19101-2, Geographic Information -- Reference model -- Part 2: Imagery, 2008.

22. Kamerman, Gary, The Infrared & Electro-Optical Systems Handbook, Volume 6, Active Electro-Optical Systems, Chapter 1. Laser Radar

23. Liadsky, Joe. Introduction to LIDAR. NPS Workshop, May 24, 2007.

24. McGlone, J. ASPRS Manual of Photogrammetry, Fifth Edition, 2004

25. Mikhail, Edward M., James S. Bethel, and J. Chris McGlone. Introduction to Modern Photogrammetry. New York: John Wiley & Sons, Inc, 2001.

26. MIL-HDBK-850, MC&G Terms Handbook, 1994

27. North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG), Air Reconnaissance Primary Imagery Data Standard, Base document STANAG 7023 Edition 3, June 29, 2005.

28. National Geospatial-Intelligence Agency. National Imagery Transmission Format Version 2.1 For The National Imagery Transmission Format Standard, MIL-STD-2500C, May 1, 2006.

29. National Imagery and Mapping Agency. System Generic Model, Part 5, Generic Sensors, December 16, 1996.

30. Open Geospatial Consortium Inc. Transducer Markup Language Implementation Specification, Version 1.0.0, OGC® 06-010r6, December 22, 2006.

31. Open Geospatial Consortium Inc. Sensor Model Language (SensorML) Implementation Specification, Version 1.0, OGC® 07-000, February 27, 2007.

32. Proceedings of the 2nd NIST LADAR Performance Evaluation Workshop – March 15 - 16, 2005, NIST 7266, National Institute of Standards and Technology, Gaithersburg, MD, March 2005.

33. Ramaswami, P. Coincidence Processing of Geiger-Mode 3D Laser.

34. Schenk, T., 2001. Modeling and Analyzing Systematic Errors in Airborne Laser Scanners. Technical Report Photogrammetry No. 19, Department of Civil and Environmental Engineering and Geodetic Science, Ohio State University.

35. Stone, W.C., *(BFRL)*, Juberts, M., Dagalakis, N., Stone, J., Gorman, J. *(MEL)* "Performance Analysis of Next-Generation LADAR for Manufacturing, Construction, and Mobility", NISTIR 7117, National Institute of Standards and Technology, Gaithersburg, MD, May 2004.

36. Wehr, A., Airborne Laser Scanning – An Introduction and Overview.

Appendix A: Exploiting Lidar "On-Demand" Metadata for Ground Covariance Generation

Use and Exploitation of the On-Demand model

This Appendix describes how a dataset containing the metadata required for the On-Demand model would be exploited to develop an absolute ground covariance matrix at a ground point as well as a full ground covariance matrix for multiple points within a dataset.

FUNDAMENTAL CONCEPTS IN ON-DEMAND EXPLOITATION

Before explaining the details of the exploitation, there are fundamental concepts related to the On-Demand model that must be discussed. The first of these is a description of what actually happens when one mensurates a point in a dataset. This process is illustrated at a high level in the diagrams below (A-1 and A-2).

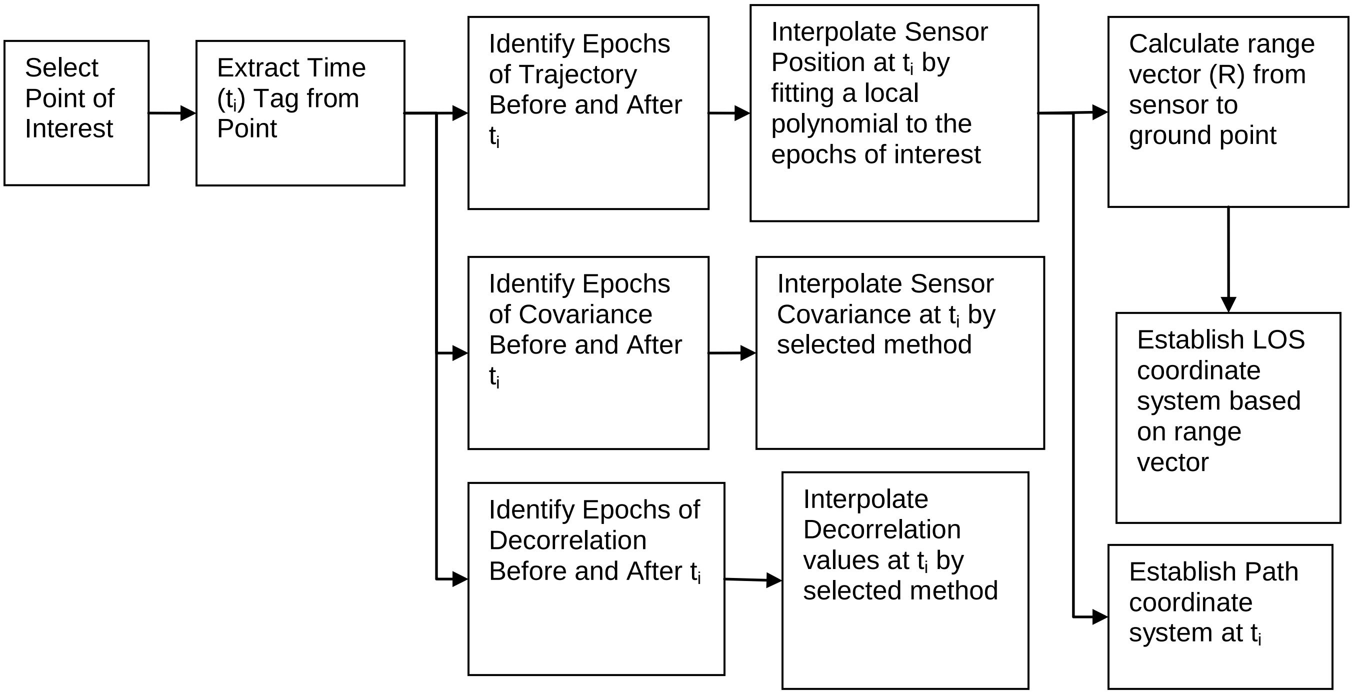
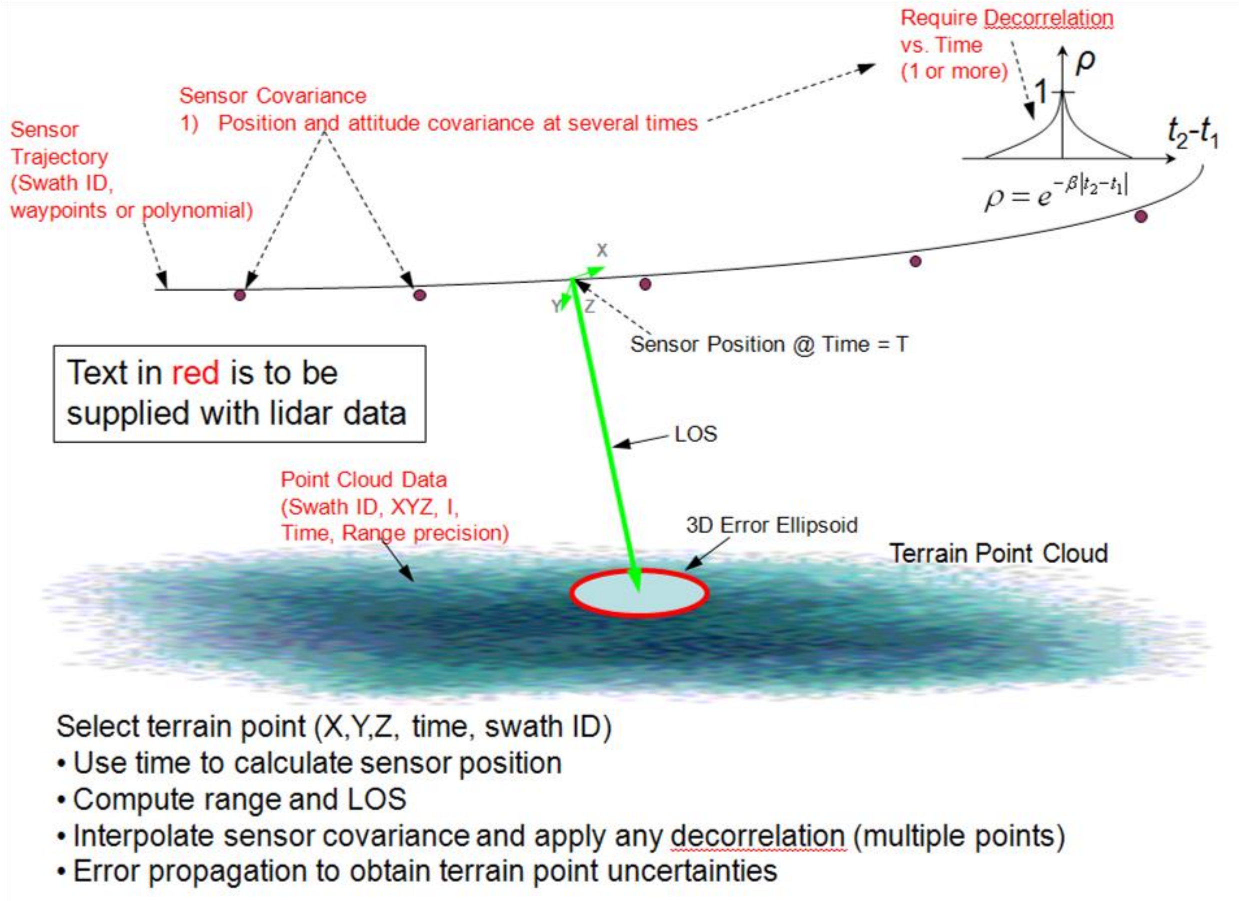

Select Point of Interest
Extract Time (t_i) Tag from Point
Identify Epochs of Trajectory Before and After t_i
Interpolate Sensor Position at t_i by fitting a local polynomial to the epochs of interest
Calculate range vector (R) from sensor to ground point
Identify Epochs of Covariance Before and After t_i
Interpolate Sensor Covariance at t_i by selected method
Establish LOS coordinate system based on range vector
Identify Epochs of Decorrelation Before and After t_i
Interpolate Decorrelation values at t_i by selected method
Establish Path coordinate system at t_i

Figure A-1. Exploitation of the On-Demand Model

These parameters, solved for using the On-Demand model, will then be used within several Community Sensor Model (CSM) functions to assist in the exploitation of the dataset. The ModelToGround() CSM function takes the current adjustable sensor parameters and applies them to the original point cloud (the model) to develop a refined model coordinate for the point of interest. Another common CSM function that is used with On-Demand data is ComputeSensorPartials. A ground point (X,Y,Z,t) is passed into the ComputeSensorPartials() function and the function passes back the partial derivatives of the ground coordinates with respect to the sensor parameters used in the model.



Text in red is to be supplied with lidar data

Select terrain point (X,Y,Z, time, swath ID)

- Use time to calculate sensor position
- Compute range and LOS
- Interpolate sensor covariance and apply any decorrelation (multiple points)
- Error propagation to obtain terrain point uncertainties

Figure A-1: Graphical View of Exploitation of the On-Demand Error Model

Absolute Error At a Point

To calculate the absolute error at a mensurated point in a point cloud, one would first follow the flow in Figure A-1. This would provide the symmetric Sensor Covariance (Σ) at the time of interest (t_i). One would then use the sensor geometry defined by the trajectory and range vector to compute the partial derivatives of the ground coordinates with respect to sensor parameters (A=dF/dS).

$$\Sigma = \begin{bmatrix} \sigma^2{}_{X_S} & \sigma_{X_S Y_S} & \sigma_{X_S Z_S} & \sigma_{X_S \omega_S} & \sigma_{X_S \phi_S} & \sigma_{X_S \kappa_S} & \sigma_{X_S R_S} \\ & \sigma^2{}_{Y_S} & \sigma_{Y_S Z_S} & \sigma_{Y_S \omega_S} & \sigma_{Y_S \phi_S} & \sigma_{Y_S \kappa_S} & \sigma_{Y_S R_S} \\ & & \sigma^2{}_{Z_S} & \sigma_{Z_S \omega_S} & \sigma_{Z_S \phi_S} & \sigma_{Z_S \kappa_S} & \sigma_{Z_S R_S} \\ & & & \sigma^2{}_{\omega_S} & \sigma_{\omega_S \phi_S} & \sigma_{\omega_S \kappa_S} & \sigma_{\omega_S R_S} \\ & & & & \sigma^2{}_{\varphi_S} & \sigma_{\phi_S \kappa_S} & \sigma_{\phi_S R_S} \\ & & & & & \sigma^2{}_{\kappa_S} & \sigma_{\kappa_S R_S} \\ Sym & & & & & & \sigma^2{}_{R_S} \end{bmatrix}$$

Eq A-1: Sensor Covariance Matrix at Time t_i

$$\underset{3\times 7}{\mathbf{A}} = \begin{bmatrix} \dfrac{\partial X}{\partial X_S} & \dfrac{\partial X}{\partial Y_S} & \dfrac{\partial X}{\partial Z_S} & \dfrac{\partial X}{\partial \omega_S} & \dfrac{\partial X}{\partial \varphi_S} & \dfrac{\partial X}{\partial \kappa_S} & \dfrac{\partial X}{\partial R_S} \\ \dfrac{\partial Y}{\partial X_S} & \dfrac{\partial Y}{\partial Y_S} & \dfrac{\partial Y}{\partial Z_S} & \dfrac{\partial Y}{\partial \omega_S} & \dfrac{\partial Y}{\partial \varphi_S} & \dfrac{\partial Y}{\partial \kappa_S} & \dfrac{\partial Y}{\partial R_S} \\ \dfrac{\partial Z}{\partial X_S} & \dfrac{\partial Z}{\partial Y_S} & \dfrac{\partial Z}{\partial Z_S} & \dfrac{\partial Z}{\partial \omega_S} & \dfrac{\partial Z}{\partial \varphi_S} & \dfrac{\partial Z}{\partial \kappa_S} & \dfrac{\partial Z}{\partial R_S} \end{bmatrix}$$

Eq A-2: Partial Derivatives of Ground Coordinates with Respect to Sensor Parameters

The covariance matrix of the ground point can then be computed as follows:

$$\Sigma_{P1} \equiv \begin{bmatrix} \sigma_X^2 & \sigma_{XY} & \sigma_{XZ} \\ & \sigma_Y^2 & \sigma_{YZ} \\ Sym & & \sigma_Z^2 \end{bmatrix} = \underset{3\times u}{\mathbf{A}}\ \underset{u\times u}{\Sigma}\ \underset{u\times 3}{\mathbf{A}^{\mathrm{T}}}$$

Eq A-3: Computing Ground Covariance Matrix at a Point

FULL COVARIANCE BETWEEN MULTIPLE POINTS

In a similar fashion, the full covariance matrix between multiple points can be calculated. However, in this case, the full sensor covariance matrix will contain a sensor covariance matrix for each point of interest (as in Eq A-1), and also the covariance information for the relationship between those points. The full covariance matrix will look as follows:

$$\underset{7n\times 7n}{\Sigma} = \begin{bmatrix} \underset{7\times 7}{\Sigma_{\mathrm{t}_1}} & \Sigma_{\mathrm{t}_1,\mathrm{t}_2} & & \Sigma_{\mathrm{t}_1,\mathrm{t}_n} \\ \Sigma_{\mathrm{t}_1,\mathrm{t}_2}^{\mathrm{T}} & \underset{7\times 7}{\Sigma_{\mathrm{t}_2}} & & \vdots \\ & & \ddots & \vdots \\ \Sigma_{\mathrm{t}_1,\mathrm{t}_n}^{\mathrm{T}} & \ldots & \cdots & \underset{7\times 7}{\Sigma_{\mathrm{t}_n}} \end{bmatrix}$$

Eq A-4: Full Sensor Parameter Covariance Matrix Between Multiple Points

The sub-matrices along the diagonal blocks are established as defined in the Figure A-2 and the section above. However, one must consider the decorrelation times to establish the sub-matrices off of the diagonal. First, the appropriate decorrelation variable (β) must be retrieved for each variable in the sensor covariance matrix. The decorrelation matrix (D_p) between the two epochs in time is then calculated as follows:

$$D_P = \begin{bmatrix} \rho_{X_S} & 0 & 0 & 0 & 0 & 0 & 0 \\ & \rho_{Y_S} & 0 & 0 & 0 & 0 & 0 \\ & & \rho_{Z_S} & 0 & 0 & 0 & 0 \\ & & & \rho_{\omega_S} & 0 & 0 & 0 \\ & & & & \rho_{\varphi_S} & 0 & 0 \\ & & & & & \rho_{\kappa_S} & 0 \\ & & & & & & \rho_{Rs} \end{bmatrix} \qquad where\ \rho = e^{-\beta|t_2 - t_1|}$$

Eq A-5: Decorrelation Matrix

It was originally proposed that the off-diagonal components of the covariance matrix between any two points in the full covariance matrix shown in Equation A-4 could be calculated by:

$$\Sigma_{t_1,t_2}(i,i) = D_\rho(i,i)\sqrt{\Sigma_{t_1}(i,i)\Sigma_{t_2}(i,i)}$$

Eq A-6: Covariance Matrix between Two Points

While this works in many situations, there were instances where numerical issues occurred when forming the ground covariance matrix. This issue would manifest itself as negative variance terms in the final ground covariance matrix shown in Equation A-4 When the covariance was exploited and a square root was taken of the negative (which should have been positive) variance term, an erroneous output such as "Not A Number" would occur and no predicted error could be determined. To avoid the potential for this issue, the proposed methodology was modified to the currently recommended method outlined below.

Now, the off-diagonal components of the covariance matrix between any two points in the full covariance matrix shown in Equation A-4 are calculated as follows. First, the decorrelations for the position, orientation, and range components are each represented by single summary decorrelation values which are calculated as follows:

$$\rho_{pos} = \frac{\rho_x + \rho_y + \rho_z}{3}, \qquad \rho_{orien} = \frac{\rho_\omega + \rho_\varphi + \rho_\kappa}{3}$$ ▯ **EMBED Equation.3** ▯▯▯

Equation A-7: Summary Decorrelation

Note that in this formulation, positional components are considered uncorrelated from the orientation and range components. Similarly, the orientation and range components are considered to be uncorrelated.

Each sensor symmetric covariance matrix for the points of interest is then divided into two symmetric submatrices, a positional component and an orientation component as shown below:

$$\Sigma = \begin{bmatrix} \sigma^2_{X_S} & \sigma_{X_S Y_S} & \sigma_{X_S Z_S} & \sigma_{X_S \omega_S} & \sigma_{X_S \phi_S} & \sigma_{X_S \kappa_S} & \sigma_{X_S R_S} \\ & \sigma^2_{Y_S} & \sigma_{Y_S Z_S} & \sigma_{Y_S \omega_S} & \sigma_{Y_S \phi_S} & \sigma_{Y_S \kappa_S} & \sigma_{Y_S R_S} \\ & & \sigma^2_{Z_S} & \sigma_{Z_S \omega_S} & \sigma_{Z_S \phi_S} & \sigma_{Z_S \kappa_S} & \sigma_{Z_S R_S} \\ & & & \sigma^2_{\omega_S} & \sigma_{\omega_S \phi_S} & \sigma_{\omega_S \kappa_S} & \sigma_{\omega_S R_S} \\ & & & & \sigma^2_{\varphi_S} & \sigma_{\phi_S \kappa_S} & \sigma_{\phi_S R_S} \\ & & & & & \sigma^2_{\kappa_S} & \sigma_{\kappa_S R_S} \\ & & & & & & \sigma^2_{R_S} \end{bmatrix} \Rightarrow \Sigma_{pos} = \begin{bmatrix} \sigma^2_{X_S} & \sigma_{X_S Z_S} & \sigma_{X_S Z_S} \\ & \sigma^2_{Y_S} & \sigma_{Y_S Z_S} \\ & & \sigma^2_{Z_S} \end{bmatrix} and\ \Sigma_{Orien} = \begin{bmatrix} \sigma^2_{\omega_S} & \sigma_{\omega_S \phi_S} & \sigma_{\omega_S \kappa_S} \\ & \sigma^2_{\varphi_S} & \sigma_{\phi_S \kappa_S} \\ & & \sigma^2_{\kappa_S} \end{bmatrix}$$

Equation A-8: Point Covariance Matrix Split into Position and Orientation Component

Singular Value Decomposition (SVD) is then performed on each symmetric submatrix to solve for the matrix square root of the covariance matrix.

$[U_{Pos\ 1}, D_{Pos\ 1}, V_{Pos1}] = SVD\ (\sum_{pos\ t1})$, $[U_{Pos\ 2}, D_{Pos\ 2}, V_{Pos\ 2}] = SVD\ (\sum_{pos\ t2})$,

And

$[U_{orien1}, D_{orien1}, V_{orien1}] = SVD\ (\sum_{orien\ t1})$, $[U_{orien2}, D_{orien2}, V_{orien2}] = SVD\ (\sum_{orien\ t2})$

Equation A-9: Singular Value Decomposition of Submatrices

Where:

U is an 3 by 3 orthonormal matrix and D is an 3 by 3 diagonal matrix.

Note that because Σ is symmetric, V = U

The square root of the each covariance submatrix can then be calculated by:

$$\Sigma^{\frac{1}{2}} = UD^{\frac{1}{2}}U^{T}$$

Equation A-10: Using the SVD results to develop the square root of the covariance

The submatrices of the covariance between times t_1 and t_2 are then formed and these submatrices are then recombined as shown below:

$$\boldsymbol{\Sigma}_{\mathbf{Pos\ t1t2}} = \boldsymbol{\rho}_{\mathbf{pos}} \left(U_{Pos1} D^{\frac{1}{2}}_{Pos1} U^{T}_{Pos1}\right)\left(U_{Pos2} D^{\frac{1}{2}}_{Pos2} U^{T}_{Pos2}\right)^{T}$$

And

$$\boldsymbol{\Sigma}_{\mathbf{Orien\ t1t2}} = \boldsymbol{\rho}_{\mathbf{Orien}} \left(U_{Orien1} D^{\frac{1}{2}}_{Orien1} U^{T}_{Orien1}\right)\left(U_{Orien2} D^{\frac{1}{2}}_{Orien2} U^{T}_{Orien2}\right)^{T}$$

And

$$\boldsymbol{\Sigma}_{\mathbf{range\ t1t2}} = \boldsymbol{\rho}_{\mathbf{range}} \sqrt{\boldsymbol{\sigma}^{2}_{\mathbf{range,t1}} \boldsymbol{\sigma}^{2}_{\mathbf{range,t2}}}$$

$$\Sigma_{t1,t2} = \begin{bmatrix} \Sigma_{\text{Pos t1t2}} & \underset{3\times3}{\mathbf{0}} & \underset{3\times1}{\mathbf{0}} \\ \underset{3\times3}{\mathbf{0}} & \Sigma_{\text{Orien t1t2}} & \underset{3\times1}{\mathbf{0}} \\ \underset{1\times3}{\mathbf{0}} & \underset{1\times3}{\mathbf{0}} & \Sigma_{\text{range t1t2}} \end{bmatrix}$$

Equation A-11: Forming the Sensor Covariance Matrix Between Times t_1 and t_2

Note in Equation A-11, the **0** represents an appropriately sized zero filled matrix so that the $\Sigma_{t1,t2}$ matrix is 7 by 7.

These submatrices are then used to populate the off-diagonal covariance elements in the full sensor covariance matrix between points of interest (Equation A-4).

The partial derivatives of the multiple ground coordinates with respect to the multiple sensor parameters (A=∂F/∂S) must again be calculated in a fashion similar to that shown above to give:

$$\underset{3n\times 7n}{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{\mathrm{nn}} \end{bmatrix}$$

where

$$\mathbf{A}_{\mathrm{ii}} = \begin{bmatrix} \frac{\partial X_i}{\partial X_{Si}} & \frac{\partial X_i}{\partial Y_{Si}} & \frac{\partial X_i}{\partial Z_{Si}} & \frac{\partial X_i}{\partial \omega_{Si}} & \frac{\partial X_i}{\partial \varphi_{Si}} & \frac{\partial X_i}{\partial \kappa_{Si}} & \frac{\partial X_i}{\partial R_{Si}} \\ \frac{\partial Y_i}{\partial X_{Si}} & \frac{\partial Y_i}{\partial Y_{Si}} & \frac{\partial Y_i}{\partial Z_{Si}} & \frac{\partial Y_i}{\partial \omega_{Si}} & \frac{\partial Y_i}{\partial \varphi_{Si}} & \frac{\partial Y_i}{\partial \kappa_{Si}} & \frac{\partial Y_i}{\partial R_{Si}} \\ \frac{\partial Z_i}{\partial X_{Si}} & \frac{\partial Z_i}{\partial Y_{Si}} & \frac{\partial Z_i}{\partial Z_{Si}} & \frac{\partial Z_i}{\partial \omega_{Si}} & \frac{\partial Z_i}{\partial \varphi_{Si}} & \frac{\partial Z_i}{\partial \kappa_{Si}} & \frac{\partial Z_i}{\partial R_{Si}} \end{bmatrix}$$

Eq A-12: Partial Derivatives of Multi-Ground Coordinates with Respect to Multi-Sensor Parameters

Note in Eq A-12, each **0** represents a 3 by 7 zero filled matrix.

The full ground covariance matrix is then established by:

$$\underset{3n\times 3n}{\Sigma}_{Ground} \equiv \begin{bmatrix} \Sigma_{\mathrm{P1}} & \Sigma_{\mathrm{P12}} & \cdots & \Sigma_{\mathrm{P1n}} \\ \Sigma^{\mathrm{T}}_{\mathrm{P12}} & \Sigma_{\mathrm{P2}} & & \vdots \\ \vdots & & \ddots & \vdots \\ \Sigma^{\mathrm{T}}_{\mathrm{P1n}} & \cdots & \cdots & \Sigma_{\mathrm{Pn}} \end{bmatrix} = \underset{3n\times 7n}{\mathbf{A}} \underset{7n\times 7n}{\Sigma} \underset{7n\times 3n}{\mathbf{A}^{\mathrm{T}}} + \underset{3n\times 3n}{\Sigma_{meas}}$$

Eq A-13: Full Ground Covariance Matrix of n Points

It should be noted that Σ_{meas} is used to incorporate any mensuration uncertainty or the uncertainty associated with selecting the points of interest in the dataset. The 7n by 7n full sensor matrix, Σ, on the right side of this equation is defined in Eq A-4.

Relative Error Between Two Points

One could use the above method to calculate the combined covariance matrix of two points (P_1 and P_2):

$$\Sigma_{\mathrm{P1,P2}} \equiv \begin{bmatrix} \Sigma_{\mathrm{P1}} & \Sigma_{\mathrm{P12}} \\ \Sigma^{\mathrm{T}}_{\mathrm{P12}} & \Sigma_{\mathrm{P2}} \end{bmatrix} = \underset{6\times 14}{\mathbf{A}} \underset{14\times 14}{\Sigma} \underset{14\times 6}{\mathbf{A}^{\mathrm{T}}} + \underset{6\times 6}{\Sigma_{meas}}$$

Eq A-14: Full Covariance Matrix Between Two Points (P1 and P2)

This ground covariance matrix for the two points could then be used to calculate the relative error covariance matrix between the two points:

$$\Sigma_{\mathrm{Relative}} \equiv \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ & \sigma_y^2 & \sigma_{yz} \\ Sym & & \sigma_z^2 \end{bmatrix} = \Sigma_{\mathrm{P1}} + \Sigma_{\mathrm{P2}} - \Sigma_{\mathrm{P12}} - \Sigma^{\mathrm{T}}_{\mathrm{P12}}$$

Eq A-15: Relative Error Covariance Matrix Between Two Points

Appendix B: Exploiting Lidar “Pre-computed” Metadata for Ground Covariance Generation

Use of Pre-Computed error model

This Appendix describes how a dataset containing the metadata required for the Pre-Computed model would be exploited to develop an absolute ground covariance matrix at a point as well as a full ground covariance matrix for multiple points within a dataset. The concept for pre-computed ground covariance can be applied in cases where the complexity of the point cloud collection and processing CONOP preclude a straight forward definition of a sensor-space projective model.

ABSOLUTE ERROR AT A POINT

Although it may be possible to store a unique covariance matrix per point, the implementation of this method may be memory intensive and require the storage of a lot of data that is very redundant. Therefore, one implementation of a pre-computed covariance model is based on the concept of covariance sets within which X, Y, Z points in the point cloud share nearly identical or very similar uncertainties in the form of 3 by 3 ground covariance matrices. Each point record in the dataset would be encoded with a covariance number that refers to a particular covariance matrix contained in a look-up table. When a point is mensurated (i), the covariance number for the point is retrieved and used as a pointer to the covariance (R) that applies for that point.

$$\Sigma_{iR} = \begin{bmatrix} \sigma_{xR}^2 & \sigma_{xy} & \sigma_{xz} \\ & \sigma_{yR}^2 & \sigma_{yz} \\ & & \sigma_{zR}^2 \end{bmatrix}$$

Eq B-1: Pre-computed Covariance Matrix at a Point

FULL COVARIANCE BETWEEN MULTIPLE POINTS

In addition to recovering the absolute error at a point, there is also the need to develop a full covariance matrix between two or more points in the dataset and also compute relative errors between points. This full ground covariance matrix would take the form:

$$\underset{3nx3n}{\Sigma}{}_{Ground} \equiv \begin{bmatrix} \Sigma_{P1} & \Sigma_{P12} & \cdots & \Sigma_{P1n} \\ \Sigma_{P12}{}^{T} & \Sigma_{P2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \Sigma_{P1n}{}^{T} & \cdots & \cdots & \Sigma_{Pn} \end{bmatrix}$$

Eq B-2: Full Ground Covariance Matrix

In the equation above, Σ_{Pn} refers to the ground covariance matrix for a specific point of interest and is obtained using the methods described above. However, Σ_{Pmn} (e.g. Σ_{P12}) represents the cross covariance between points and the concept of decorrelation is used to calculate it. First, a correlation factor is determined between two points (P1 and P2) as follows:

$$\rho_{12} = e^{-\beta_{\mathrm{U}}\delta_{\mathrm{U}}} * e^{-\beta_{\mathrm{V}}\delta_{\mathrm{V}}} * e^{-\beta_{\mathrm{W}}\delta_{\mathrm{W}}}$$

Eq B-3: Correlation between Points

In Eq B-3, δ_U, δ_V, δ_W represent components of the vector between the two points. The β_U, β_V, β_W are the exponential coefficients that control how the correlation between points varies with their separation. For cases under consideration here, W is most closely associated with the lidar LOS range dimension. It is assumed that in most instances β_W will be zero. The ρ_{12} represents the decorrelation between the covariance matrices of points A and B based on their separation in the U, V, and W directions; in this formulation their values are restricted between 0 and 1. The U, V, and W space is defined to account for a preferred decorrelation direction (resulting from data collection geometry, for example) that may not be parallel to the primary X, Y, and Z coordinate axes.

Three angles θ_1, θ_2 and θ_3 are defined to permit the transformation of points from the X, Y, Z frame to the U, V, W frame. These angles are defined in Appendix C.

The cross-covariance matrix between any two points (P1 and P2) is then calculated by Eq B-4. Note that the matrix square-root is accomplished using SVD (see Appendix A).

$$\Sigma_{P1,2} = \rho_{P12}\sqrt{\Sigma_{P1}\Sigma_{P2}}$$

Eq B- 4: Cross covariance matrix between two points, A and B.

Once the full covariance matrix is obtained, the relative error covariance matrix between points can then be calculated by:

$$\Sigma_{\mathrm{Re}\,lative} = \Sigma_{P1} + \Sigma_{P2} - \Sigma_{P12} - \Sigma_{P12}^{T}$$

Eq B- 5: Relative covariance matrix between two points, P1 and P2.

Appendix C: Coordinate System Transformations

A. Coordinate System Transformation

Alignment of coordinate systems is accomplished via translations and reorientations through rotations. Translating between different references is a simple linear shift in each axis; *x, y, and z*. Axis alignment, or making the axis of each system parallel, is accomplished by three angular rotations as described below.

Beginning with a coordinate system defined by (*x,y,z*), the first rotation will be about the x-axis by angle θ_1 (i.e., positive *y*-axis rotates toward the positive z-axis), see Figure C-1. The resulting orientation will be designated (x_1,y_1,z_1).

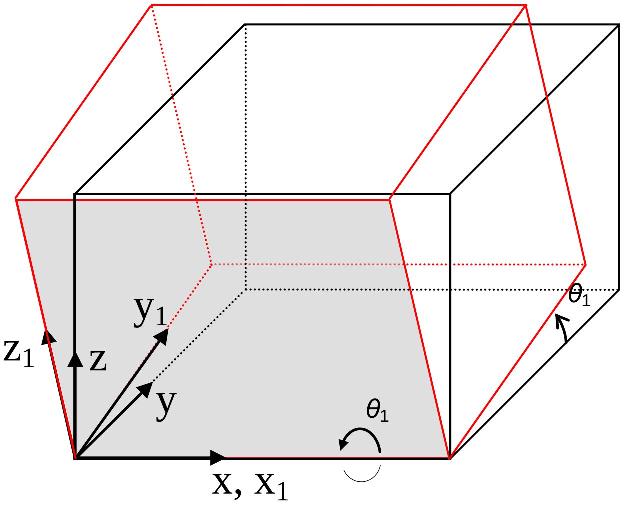


Figure C-1: First of three coordinate system rotations

The second rotation will be by angle θ_2 about the once rotated *y*-axis (positive z_1-axis rotates toward the positive x_1-axis), see Figure C-2. The resulting orientation will be designated, (x_2,y_2,z_2).

The final rotation will be by angle θ_3 about the twice rotated *z*-axis (positive x_2-axis rotates toward the positive y_2-axis), see Figure C-3. The resulting orientation will be designated, (x_3,y_3,z_3).

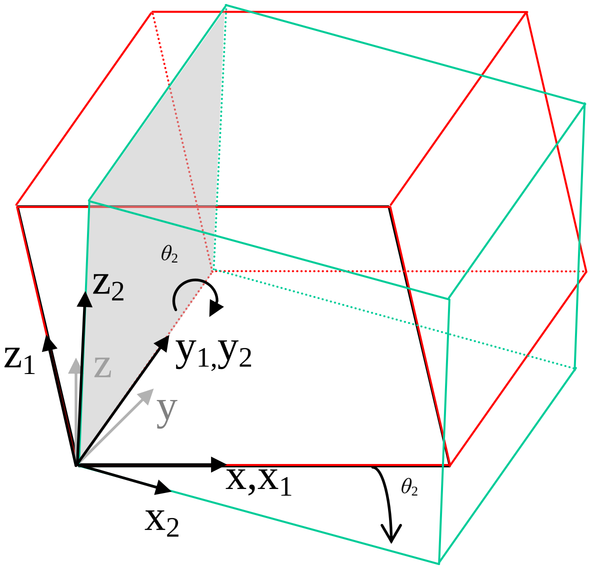


Figure C-2: Second of three coordinate system rotations

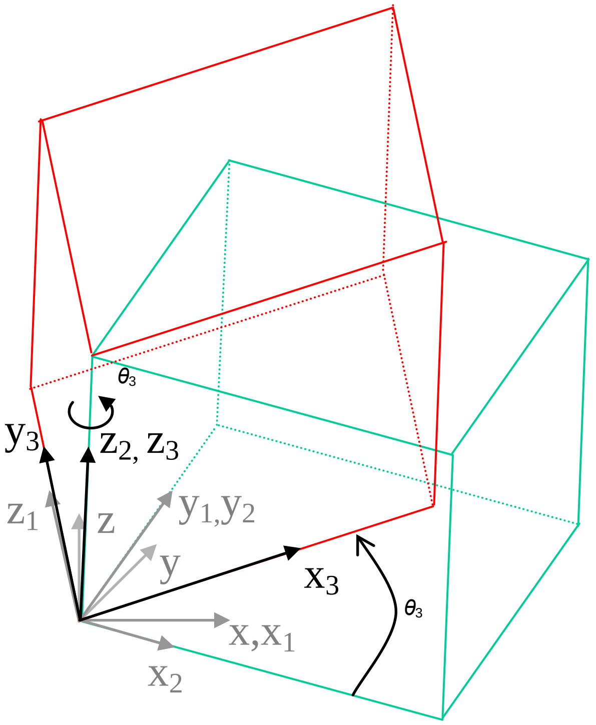


Figure C-3: Last of three coordinate system rotations

The resulting transformation matrix **M**, as, for example that is given in Eq. 49 represents the orientation of one three-dimensional (3D) coordinate system with respect to another 3D system. In this case it represents the change in orientation of the initial system, here designated by x_1,y_1,z_1, to make it transform to the final system, x_3,y_3,z_3. If the two systems related by **M** had a common origin, then **M** would be all that is needed to **transform** the coordinates with respect to x_1,y_1,z_1, to coordinates with respect to x_3,y_3,z_3 (by simply premultiplying the former by **M** to get the latter). In most situations, the coordinate systems do not have the same origin, then the **transformation** from one to the other will involve translation in addition to rotation. We have two possibilities: either rotating first then translating, or translating first to make the two systems have the same origin then rotating.

Matters become somewhat complicated when we have to deal with more than three systems of coordinates which are not translations of each other and do not have a common origin. In these situations, one has to be careful as to the rotation matrices and translation vectors to use. As an illustration, we use a simplified two-dimensional example in order to demonstrate the sequencing requirements. Beginning with a coordinate system defined by (x_1,y_1), we desire a transformation to a third coordinate system (x_3,y_3), via an intermediate coordinate system (x_2,y_2), see Figure C-4.

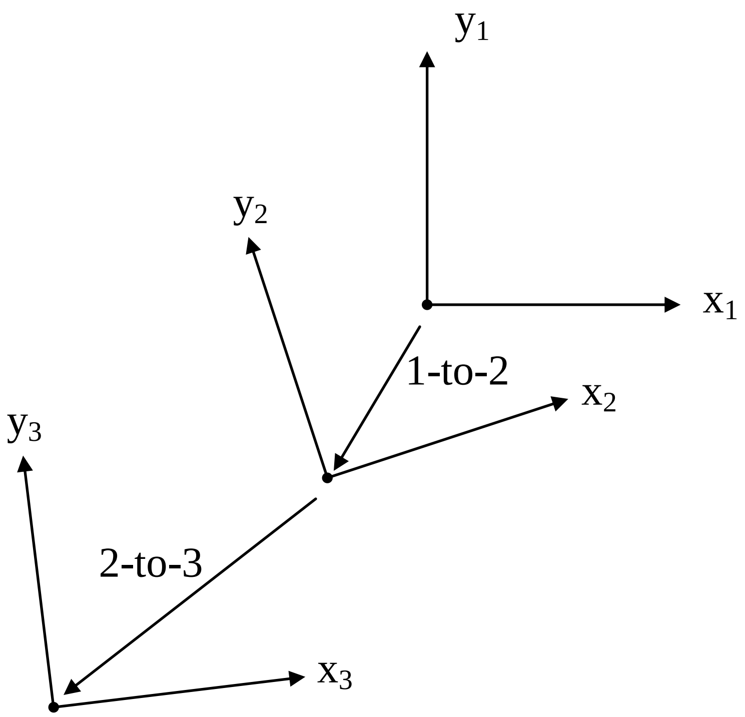


Figure C-4: Coordinate system transformation example

The first step is to transform from the initial reference to the second. The option is to rotate first and then translate, or vice versa; but to be consistent throughout. We chose to rotate first, and then translate. Therefore, transformation from the first frame to the second is illustrated by Figure C-5.

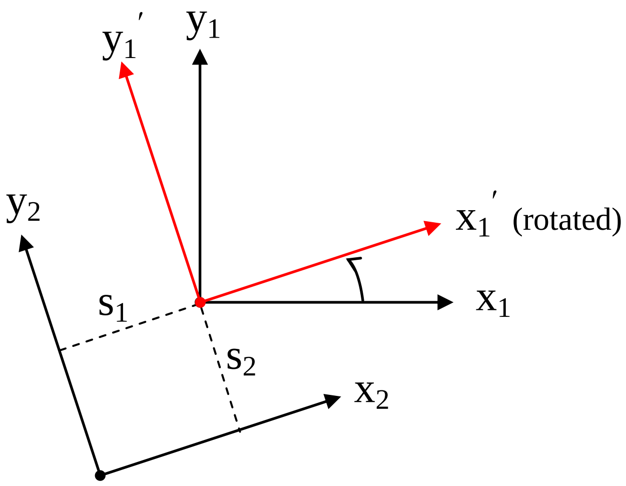


Figure C-5: First of two coordinate system transformations

The new orientation may now be defined by the following equation.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \mathbf{M}_{1-2} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

Eq. C-1

where **M**$_{1\text{-}2}$ is the rotation matrix that rotates frame one to frame two, and s_1 *and* s_2 define the translations along x_1*'* and y_1*'* (or x_2 and y_2), respectively, to effect a common origin.

Similarly, the process for transforming from the second to the third frame is as shown in Figure C-6.

This transformation may be defined by the following equation.

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \mathbf{M}_{2-3} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} s_1' \\ s_2' \end{bmatrix}$$

Eq. C-2

where **M**$_{2\text{-}3}$ is the rotation matrix from frame two to frame three, and s_1*' and* s_2*'* define the translations along x_2' and y_2' (or x_3 and y_3), respectively.

The transformations above may be combined into a single equation as follows:

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \mathbf{M}_{2-3} \left\{ \mathbf{M}_{1-2} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \right\} + \begin{bmatrix} s_1' \\ s_2' \end{bmatrix}$$

Eq. C-3

Although more complex, a similar process is applied for a 3D transformation, as is needed for sensor modeling purposes. In those cases, more intermediate transformations are likely to be necessary, particularly to account for multiple gimbals.

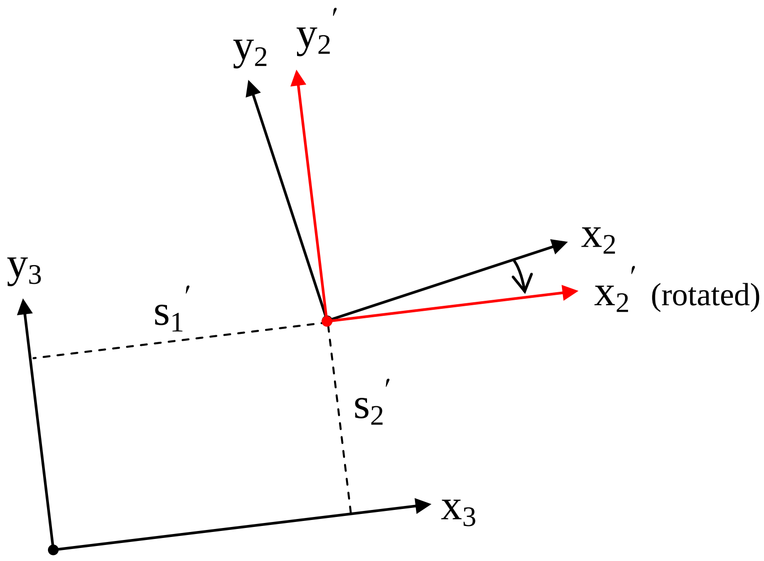


Figure C-6: Last of two coordinate system transformations